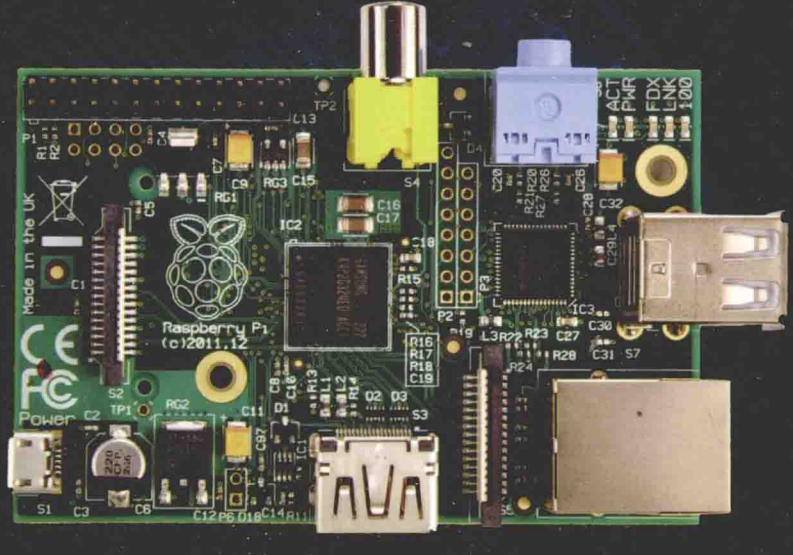
Apress®

树莓派学习指南

(基于Linux)

[英] Peter Membrey
[澳] David Hows 著
张志博 孙峻文 译

Learn Raspberry Pi with Linux



人民邮电出版社
POSTS & TELECOM PRESS

树莓派学习指南

(基于Linux)

本书是树莓派入门参考书的明智选择。两位Linux专家将指导你学习如何进行树莓派的安装配置，包括输入输出设备、网络连接、桌面设置等等。你还将学会如何驾驭命令行工具，从而迈出成为树莓派专家的第一步。通过对不同Shell以及各种命令的学习，你将登堂入室成为高级用户。最终，你可以用树莓派创建各种实用应用，诸如网页应用、家庭监控应用、媒体中心等。

通过本书，你将学会：

- 定制树莓派的桌面环境；
- 各种实用命令；
- 如何进行网络连接；
- 如何让树莓派摆脱线缆束缚；
- 如何让树莓派变身为Web服务器、家庭监控应用以及媒体中心。

树莓派的优秀毋庸置疑，Linux更是让它如虎添翼。本书将从零开始，教会你如何解放它的潜力，实现各种精彩的应用。

装帧设计：任文杰

分类建议：计算机／程序设计
人民邮电出版社网址：www.ptpress.com.cn

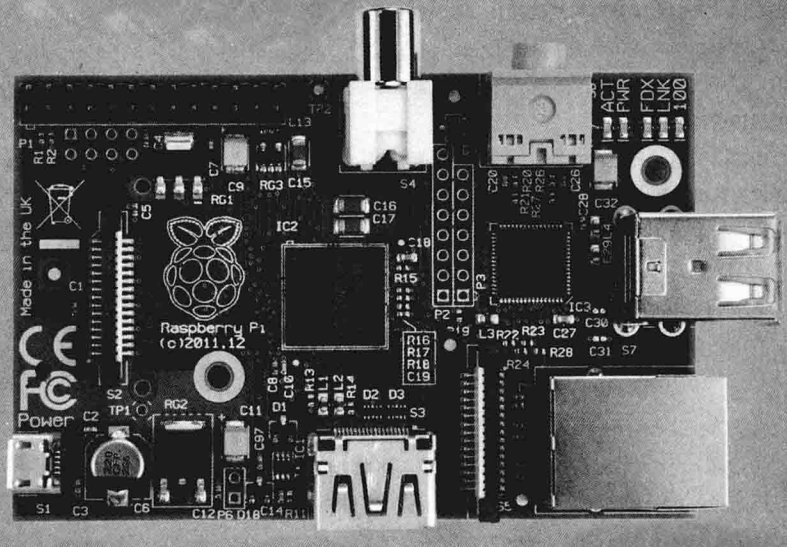
ISBN 978-7-115-34497-7

ISBN 978-7-115-34497-7
定价：49.00 元

树莓派学习指南

（基于Linux）

[英] Peter Membrey
[澳] David Hows 著
张志博 孙峻文 译



人民邮电出版社
北京

图书在版编目（CIP）数据

树莓派学习指南 : 基于Linux / (英) 门伯里(Membrey,P.), (澳) 豪斯 (Hows,D.) 著 ; 张志博, 孙峻文译. -- 北京 : 人民邮电出版社, 2014.4
ISBN 978-7-115-34497-7

Ⅰ. ①树… Ⅱ. ①门… ②豪… ③张… ④孙… Ⅲ. ①Linux操作系统－指南 Ⅳ. ①TP316.89-62

中国版本图书馆CIP数据核字(2014)第029013号

版 权 声 明

Learn Raspberry Pi with Linux
By Peter Membrey, David Hows, ISBN: 978-1-4302-4821-7
Original English language edition published by Apress Media.
Copyright ©2013 by Apress Media.
Simplified Chinese-language edition copyright ©2014 by Post & Telecom Press
All rights reserved.
本书中文简体字版由 Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。
版权所有，侵权必究。

内 容 提 要

树莓派（Raspberry Pi）是一款基于 Linux 系统的、只有一张信用卡大小的卡片式计算机。由于功能强大、性能出色、价格便宜等特点，树莓派得到了计算机硬件爱好者以及教育界的欢迎，风靡一时。

本书是学习在树莓派上基于 Linux 进行开发的一本实践指南。全书共 3 个部分 11 章，第一部分是前两章，讲述如何设置和运行图形用户界面（GUI）。第二部分是第 3 章到第 7 章，讲述如何驾驭命令行。第三部分是第 8 章到第 11 章，带领读者进行更多有趣的探索，以玩转树莓派。

本书覆盖了树莓派的许多主题，适合程序员、计算机软硬件爱好者，以及对树莓派感兴趣的读者阅读，也适合作为树莓派相关实践课程的基础教程。

◆ 著　[英] Peter Membrey　[澳] David Hows
译　张志博　孙峻文
责任编辑　陈冀康
责任印制　程彦红　焦志炜

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路 11 号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

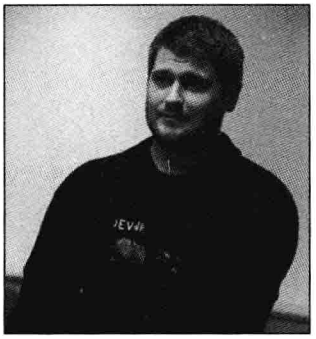
◆ 开本：800×1000　1/16
印张：13.75
字数：313 千字　2014 年 4 月第 1 版
印数：1－3 000 册　2014 年 4 月河北第 1 次印刷

著作权合同登记号　图字：01-2013-8454 号

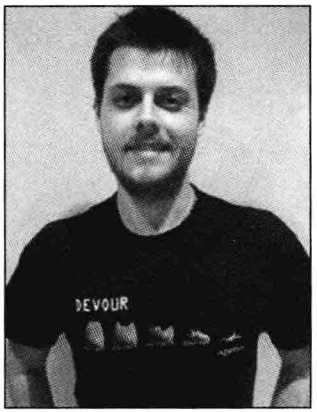
定价：49.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315

作者简介



Peter Membrey，资深软件工程师，17 岁获得 RHCE 认证，近15年以来一直致力于用Linux和开源解决方案解决各类工程问题，曾就职于 Red Hat，并撰写过多本开源软件相关书籍，硕士毕业于英国利物浦大学信息安全专业，目前在香港理工大学攻读博士学位，研究兴趣包括云计算、大数据安全。现与妻子 Sarah 和儿子 Kaydyn 定居于香港，让他颇为自嘲的是近年来他的广东话水平不进反退。



David Hows，毕业于澳大利亚新南威尔士的卧龙岗大学。儿时他就热衷于优化家用电脑的性能，从那以后便一发不可收拾。他的足迹踏遍系统管理员、软件开发者、解决方案架构师和数据库工程师等各种岗位。多年来他一直妄图进军足球圈而不成，因此他的咖啡杯上写着这样几个字：满腹牢骚。

目　录

第 1 章　初尝树莓派 …… 1

1.1　新出炉的派 …… 1
1.2　树莓派配料表 …… 3
1.3　Linux …… 7
1.4　将 Raspbian 烧录到 SD 卡 …… 8
1.5　首次启动 …… 14
1.6　配置树莓派 …… 15
本章小结 …… 23

第 2 章　开始探索 …… 24

2.1　欢迎使用 LXDE …… 24
2.2　开始菜单 …… 27
本章小结 …… 40

第 3 章　驾驭命令行 …… 41

3.1　曾经的计算机 …… 42
3.2　3 种终端 …… 45
3.3　欢迎来到命令行的世界 …… 51
3.4　不同的 Shell …… 53
本章小结 …… 53

第 4 章　玩转文件系统 …… 54

4.1　什么是文件系统 …… 54
4.2　万物皆文件 …… 56
4.3　文件系统结构 …… 58
4.4　让我们来实践一下 …… 60
4.5　用户和用户组 …… 65
4.6　文件权限 …… 66
4.7　软链接和硬链接 …… 70
本章小结 …… 71

第 5 章　重要命令 …… 72

5.1　掌控全局 …… 72
5.2　快去读那该死的文档 …… 73
5.3　系统资源监控 …… 74
5.4　文件命令 …… 82
本章小结 …… 89

第 6 章　在命令行中编辑文件 …… 90

6.1　什么是文本文件 …… 90
6.2　文本编辑器的竞争 …… 91
6.3　使用 nano …… 92
6.4　使用 Vim …… 97
6.5　Vim Style 的复制、粘贴 …… 104
本章小结 …… 105

第 7 章　管理你的树莓派 …… 106

7.1　远程访问树莓派 …… 106
7.2　通过 BASH 学编程 …… 107
7.3　第一个 BASH 脚本 …… 116
7.4　安全和用户管理 …… 121
本章小结 …… 122

第 8 章　搭建 LAMP …… 123

8.1　第一步 …… 123

8.2 L：Linux 系统……124
8.3 A：Apache 服务器……124
8.4 M：MySQL……129
8.5 P：PHP……138
8.6 实现一个简易网页应用……141
本章小结……150

第 9 章 无线树莓派……151

9.1 一点准备工作……152
9.2 快刀斩乱麻……152
9.3 为树莓派添加 WiFi……155
9.4 移除电源线……162
9.5 外置电源的 USB Hub……163
本章小结……164

第 10 章 树莓监控派……165

10.1 需要的原料……165
10.2 准备……166
10.3 开始动手……166
10.4 运动捕捉……170
10.5 自动报警……174
10.6 模块拼接……178
10.7 开发路线图……184
本章小结……185

第 11 章 树莓派媒体中心……186

11.1 使用 XBMC 播放视频……186
11.2 开始使用 XBMC……193
11.3 使用 AirPlay 播放音乐……196
本章小结……201

第 1 章

初尝树莓派

从本章开始，我们正式进入折腾环节。需要提醒的是，如果你迫不及待想让树莓派运行起来，跳过序言也并无不可，关键的内容在本章中都会涉及。但我们希望你能够记得找时间阅读序言部分，这样你才会明白树莓派的由来以及它独一无二的原因。

好了，让我们开始折腾吧。从开箱开始，我们会涉及各种如何使树莓派真正运行起来的内容。连接好各种接线后，我们会动手安装并设置 Raspbian Linux。你可能会发现有些配置方式比较陌生，别担心，在我们的帮助下你一定可以顺利完成。结束这一章之前，我们还会初步介绍 Raspbian 桌面（第 2 章会进一步展开这部分内容）。

1.1 新出炉的派

从快递员手中接过你期待已久的包裹，迫不及待地撕开牛皮纸信封后，你会发现一个小盒子（真的很小）。或许你会忍不住向信封里再张望一下，看看是不是还有些别的什么。别怀疑，亲爱的读者，订购树莓派后你会收到的就只有这样一个小盒子（见图 1-1）。

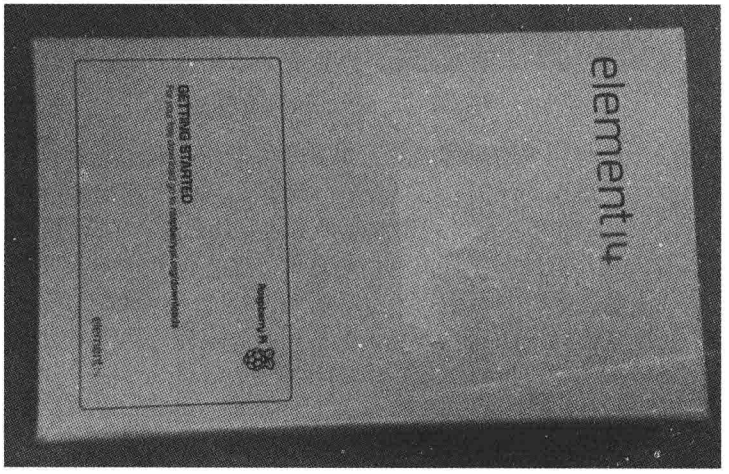


图 1-1 包裹里的所有内容——一个小盒子

为什么要看这张盒子的照片呢?

首先是告诉大家快递没出问题，包裹里确实只有这个小盒子。还有就是告诉大家，想要让树莓派真正运行起来，仅凭这个小盒子里的内容是不够的，我们还需要些小零件。别担心，下面我们会一一介绍。

你可能会注意到盒子上用粗体字写着“GETTING STARTED”（从这里开始），后面还有一个树莓派网站的网址。如果你先浏览该网址而不是急着拆开包装的话，你会发现启动树莓派没有什么难度。但或许你不相信，当我们最初拿到树莓派时，因为忙着向大家献宝，所以完全没有注意如此重要的信息。无独有偶，就我们所知有个树莓派玩家物尽其用地将包装盒改装成支架使用，却对包装上的提示视而不见。

我们还是从 http://raspberrypi.org/downloads/开始旅程。如果你还没有订购树莓派（或是还没拿到手），可以看一下图 1-2（树莓派和鼠标的合影），通过它你可以对树莓派的大小有个感性的认识。

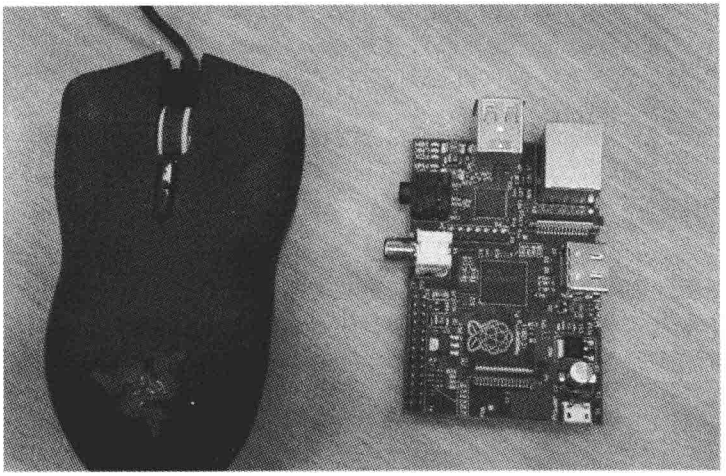


图 1-2　树莓派与一只普通鼠标的大小比较

即便你对树莓派的小巧身材早有耳闻，真正拿在手上依旧会让你眼前一亮。

我们拿到了树莓派，不过到底还需要哪些配件才能让它真正工作起来，我们也不是很确定。虽然我们也在计算机行业工作多年，但是树莓派是不是需要什么特殊的配件？只有用起来才知道。

如果你不是拥有超大工作室的科学怪人，也不是常年捣弄计算机的极客，那你可能需要购置些必要的配件。幸运的是这些配件并不难找，电脑卖场里都能买到所需的配件。如果你讨厌寻寻觅觅，有不少公司为初学者提供了工具箱，包括了各种必要配件。配件的更新换代很快，最好的办法是在网上搜索“Raspberry Pi Starter Kit”（树莓派新手套件）。

注意 虽然在本书中我们选择使用 HDMI 作为视频输出接口，但你也可以选择 DVI、复合视频输出端子等其他接口。

1.2 树莓派配料表

为了完美烘培（使用）树莓派，需要如下配料（配件）：

- 树莓派
- Micro USB 线缆
- USB 电源适配器
- HDMI 线缆（A 型）
- 支持 HDMI 的显示设备
- SD 卡：2-64GB（推荐配置至少 8GB）
- SD 卡读卡器
- USB 键盘、鼠标

Micro USB 线缆

如果有 Android 手机或是 Kindle，这些设备上的线缆很可能可以用在树莓派上。虽然也可以将 USB 线连接在电脑的 USB 接口上，但树莓派上的接口不是用来传输数据的（只作为电源使用）。USB 接口的种类繁多，如图 1-3 所示。

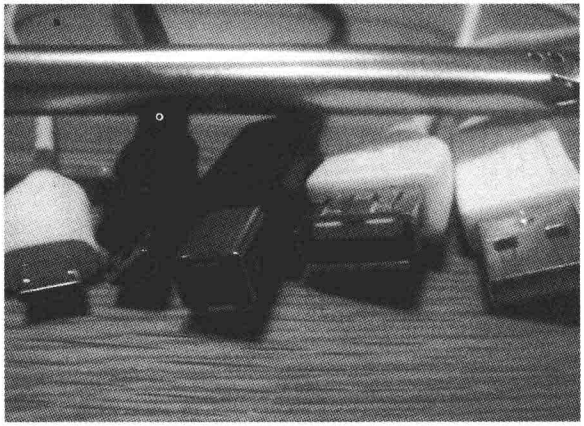


图 1-3 各种类型的 USB 接口

这里我们要用到的是最左边的接口，它被称为 Micro USB。需要注意的是，Micro USB 接口与 Mini USB 接口（图 1-3 中左二）十分容易混淆。因为搞错了而多跑一趟商店可不是什么好主意。

USB 电源适配器

电源适配器实际上不是必选配件。电脑上的 USB 接口可以为 A 型树莓派（没有板载以太网口）提供足够的电力，对于 B 型则无法保证。当然，这样的话你就不得不把树莓派放在电脑旁边了，这或许不是你最终想要的。保证树莓派拥有独立的电源适配器也可以确保当你想移动电脑时不会为难。

幸运的是，可以为树莓派供电的 USB 电源适配器并不难找。只是有一点需要注意，B 型树莓派需要 700 毫安以上的电流输入。为安全起见，你不妨找一个提供 1000 毫安电流输出的适配器。根据我们对适配器的调查，1000 毫安电流输出的适配器并不难找。我们也发现了一些 500 毫安的适配器，不过它们并不满足树莓派的需求。

HDMI 线缆

近年来，HDMI 接口成为了连接显示器和电视的事实标准。这意味着如果一台设备支持 HDMI 接口，为它找台显示器是不难的。现在听起来这或许没什么稀奇，但几年前显示接口十分混乱。那时，无论是把电脑连接到电视上还是把 VCR 连接到显示器上都令人抓狂（有趣的是，如果时间倒退到 20 年前，所有的家用电脑都是连接到电视上的，比如序言中提过的 Commodore 64 和 Spectrum）。当然可以通过转接器解决这个问题，也可以选择有各种接口的高端设备，但那是另一个问题了。

幸运的是，树莓派提供了 HDMI 接口，我们可以忘却各种接口带来的烦恼。显示器和电脑上都是 A 型 HDMI 接口，只需要找一条这种接口的 HDMI 线缆把它们连接就可以了。A 型 HDMI 接口应用广泛，如果有台 Xbox 360 或是带 HDMI 接口的其他设备，那你八成不用额外再买别的线缆了。

支持 HDMI 的显示设备

不出意外的是需要支持 HDMI 接口的显示设备。别担心，近年来生产的电视基本都有一个或一个以上的 HDMI 接口。实际上，不少电视提供了 3 个或 4 个接口，方便连接各种新型数码设备。

判断一台显示设备是否支持 HDMI 的方法非常简单。直接找 HDMI 接口就可以了。在电视上找到这种接口的概率相对于显示器要高一些。

我们当时遇到的情况是，旧的显示器不支持 HDMI，卧室的电视支持。我们不想占用电视，所以我们买了台支持 HDMI 的新显示器。遇到这种情况也可以买一个 DVI-HDMI 转接器来解决，我们只是不想放过购置新玩具的机会。前面提过，树莓派还支持其他的显示方案，所以除了 HDMI 也有其他选择，比如复合视频输出端子。

SD 卡

多数电脑使用硬盘作为主要存储设备，但即便是笔记本硬盘的体积也比整个树莓派还要大。虽然新兴的固态硬盘（SSD）以低能耗著称，但相对于树莓派来说它的能耗还是太高了。幸运的是，我们还有其他选择。这种技术数码相机已经使用了很多年，即 SD 卡。虽然在空间大小和读写速度上与硬盘相比存在差距，但 SD 卡在能耗和体积上比硬盘都小得多。64GB 的 SD 卡（树莓派支持的最大大小）对于树莓派来说显然绰绰有余。

虽然大部分 SD 卡都适用[1]，但是我们建议尽可能购买高速 SD 卡（Class 10 是最佳选择）。选

[1] 译者注：树莓派最低要求 Class 4。

择相对知名品牌的 SD 卡可以帮你规避可能存在的兼容性问题。

现在市面上的闪存卡种类繁多，图 1-4 所示的是我们所需的 SD 卡。

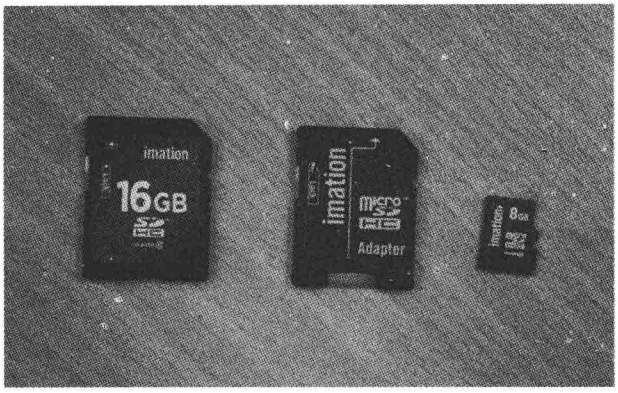


图 1-4　SD 卡和 Micro SD 卡、卡套

SD 卡读卡器

显然，现在你需要一种读写 SD 卡的方法。树莓派内置了 SD 卡槽，但它不具有其他存储设备。所以为了将 Raspbian Linux 安装到 SD 卡上，需要将 SD 卡读卡器连接到电脑上进行操作。你也许会想，如果 SD 卡上有 Linux 系统，就可以通过它启动树莓派，但是如果想往该 SD 卡上写入 Raspbian Linux，则需要抹掉已有的 Linux 系统。听起来有点像绕口令，总之还是在电脑上操作比较方便。

随着数码相机的普及，越来越多的电脑内置了读卡器。有些电脑上还内置了多合一读卡器，可以读写各种闪存卡。这样看来，找个 SD 卡读卡器也不是什么难事。如果你的电脑上没有，也可以在电脑卖场很容易地买到支持 SD 卡的外置读卡器。外置读卡器一般都很便宜，并且支持多种闪存卡。图 1-5 所示的就是一个外置读卡器。



图 1-5　外置读卡器

USB 键盘、USB 鼠标

别忘了还需要键盘、鼠标。树莓派不会读心术，所以还是需要通过键鼠控制它。需要注意的是部分无线键鼠[1]可能不适用。

搞定！

我们已经一切准备就绪，将各种设备连接到树莓派上十分容易。这些接口设计得很好，不存在误插的可能，如图 1-6 所示。

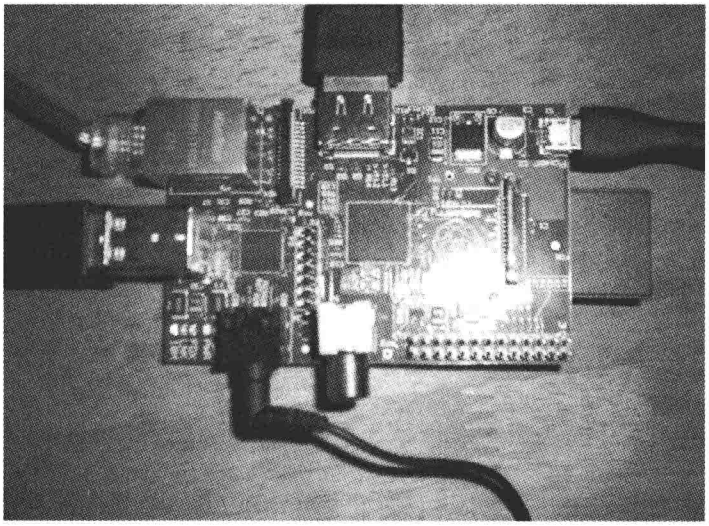


图 1-6　准备就绪的树莓派

将所有线缆连好后，插上电源，打开显示器，然后你会看到的是……铛铛铛铛……一片黑屏。

别惊慌

当电脑启动时，BIOS 会进行加电自检——测试内存、启动显示、硬件初始化。部分电脑（主要是苹果公司的产品）会使用 EFI 来执行此类操作。不过站在用户的角度来看，这两者间并没有太大差别。本质上它们都是负责加电自检并启动操作系统的。

BIOS 天生是种很吵的程序，如果遇到了错误你会听到各种蜂鸣声（如果想明白不同蜂鸣声的含义，就不得不去翻找早不知丢到哪里的主板说明书了），还会看到各种复杂的报错信息。不过，至少这些报错告诉你电脑还能点亮。而树莓派在没有可启动的操作系统时屏幕上不会显示任何内容。如果期待诸如蓝天白云之类的欢迎画面，你可能要失望了。

现在你已经知道黑屏的原因了。是时候进入下一阶段了，让我们来学习如何安装 Linux 吧。

---

[1] 译者注：指不带 USB 接收器的那些，比如 Apple Magic Mouse。

1.3 Linux

作为一本初学者指南，我们会简要介绍 Linux 的由来、特点以及一些初学者常犯的错误。如果你对这部分内容有所了解或不太感兴趣，不妨跳过本节，继续下一小节“下载 Raspbian”。

Linux 是什么？

解释 Linux 是什么这个问题即便对专家来说也会感到困扰。原因在于 Linux 这一概念在不同语境下代表的含义相差很多。当谈到操作系统时，我们可以很容易理解 Microsoft Windows 和 Apple Mac OS X 是完全不同的东西。当我们说“我在使用 Windows”不会产生歧义，而 Linux 的情况就不是这样了。

Linux 只是一种操作系统内核。这意味着它主要处理系统底层操作，诸如设备驱动、网络访问和磁盘读写。人们真正接触的是运行在 Linux 上的软件。让人更困扰的是，人们往往对同类软件的选择存在较大的分歧。

由于 Linux 是开源的，任何人都可以折腾出自己的发行版。“龙生九子，九子各不同”。各发行版都使用内核，但为满足不同需求，各自捆绑了多种多样的应用程序。不同的 Linux 发行版都有自己的特点：Red Hat Enterprise Linux（RHEL）健壮，稳定，并且提供优秀的支持；Fedora 发布速度快（每 6 个月左右发布一次）并且更新迅速；Gentoo 通过让用户自己编译系统的方式，让系统尽可能地最优化运行；Debian 则在安全稳定和引入新特性间保持平衡。

“开源”是什么意思

在一台计算机还价值连城的时代，计算机的软硬件作为一个整体进行销售，没有人单独销售软件。那时软件公司必须搭配硬件去推销他们的产品。软件公司一直着力于保护自己的产品，其中的一种做法是只提供可执行文件而不提供源代码，这也意味着用户无法根据自己的需求对软件进行修改或增强。

所以有些人认为软件应该总是和源代码一起发布，这样用户才能够按需修改。开源意味着人们可以自由地交换、修正和改进软件。

这个主题或许要另写一本书才讲得清楚，不过你不妨看看作者去年就这一主题做的一次演讲：http://www.youtube.com/watch?v=c-1LQIGh6cI[1]。

你或许想问，到底哪个发行版最好呢？这里需要注意的是，根本不存在完美的发行版。我们需要根据不同的需求进行选择。就树莓派来说，Raspbian（基于 Debian）是官方支持的发行版，它简单易用，支持良好和更新迅速。本书将主要讲述这一发行版。如果你喜欢与众不同，Apress 出版社出版的 Brendan Horan 所著的《Practical Raspberry Pi》会教你如何安装 Fedora，甚至如何自己编译 Gentoo。

Linux 内核在 20 世纪 90 年代初第一次亮相时，没人会想到它在计算机领域引起一场地震。

[1] 译者注：由于众所周知的原因，这一网址可能无法访问。

其实在 Linux 内核发布前，开源软件运动已经持续了很久，期间诞生了无数优秀的 UNIX 程序（诸如 GCC 编译器）。但这些程序却只能运行在私有操作系统上。为了契合开源精神，开发开源操作系统内核的呼声日渐高涨。Linux 就是在这样的背景下诞生的。Linux 内核为计算机领域带来的自由和希望是一个很大的主题，我们所知道的是 Linux 内核真正将开源运动引入了公众视野。

那我们到底能用 Linux 做些什么呢？这里不妨先卖个关子，可以说唯一的束缚是我们的想象力。千里之行，始于足下。让我们先从安装它开始吧。

下载 Raspbian

首先，从树莓派下载页 http://raspberrypi.org/downloads/下载 Raspbian。在这个网页里除了 Raspbian，我们还能看到一些其他发行版的下载链接。想知道为什么我们不选择它们，读读“为什么选择 Raspbian”这部分吧。

为什么选择 Raspbian

如果你以前接触过 Linux，你可能会想为树莓派安装 Raspbian 以外的其他发行版。不过大多数人会在树莓派上运行 Raspbian 这一官方支持的发行版。它有着最好的支持，大多数的教程（包括本书）都是针对它编写的。这意味着选择其他发行版需要承担缺少相应支持的风险。

此外，树莓派基于 ARM 构建，因此它与你的电脑有些不同。比如，它没有 PCI 总线，所以依赖 PCI 的工具并不适用。这意味着除官方支持的 Raspbian，其他发行版存在潜在风险。简而言之，Raspbian 是最适合初学者的选择。

Linux 发行版以往总是以 CD 或 DVD 等光盘为介质发布。从网上下载的光盘镜像文件的内容与这些光盘并无二致。这样就可以将镜像文件烧录到光盘上进行安装。树莓派的情况稍有不同，它并没有内置光驱。为了解决这一问题，树莓派基金会为我们准备了已经配置好的 SD 卡镜像文件。

1.4　将 Raspbian 烧录到 SD 卡

将 Raspbian 烧录到 SD 卡与简单的文件复制稍有不同。我们下载的镜像文件是一份完整的 SD 卡内容克隆。为了完成烧录的工作，我们需要使用前面提到的一些工具。

注意 你所下载的 SD 卡镜像文件被打包在一个 zip 压缩包里。在 Windows 7 下，双击即可以文件夹的形式打开 zip 压缩包。将 SD 卡镜像文件复制到自己的文件夹下以备使用。在 Mac OS X 下，双击 zip 压缩包后 SD 卡镜像文件将会自动被解压缩到“下载”（Downloads）目录中。

使用 Image Writer 进行烧录（Windows）

在 Windows 下推荐使用 Image Writer 这一工具进行烧录，你可以在树莓派下载页上找到它的

链接，我们需要的是如图 1-7 所示的 zip 压缩包。

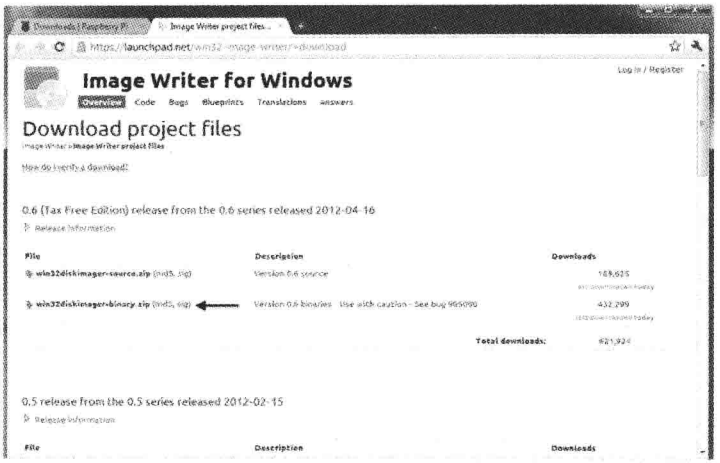


图 1-7　下载 Image Writer

将 zip 压缩包解压到任意文件夹，双击其中的 Win32DiskImager，你会看到如图 1-8 所示的警告信息。

1．单击 Extract All（提取所有文件），在弹出的对话框中记得勾选“Show Extracted Files When Complete”（完成时显示提取的文件）。

2．单击 Extract（提取），稍等片刻后你就会看到解压后的文件夹了。

3．双击 Win32DiskImager 程序，你会看到如图 1-8 所示的警告信息。

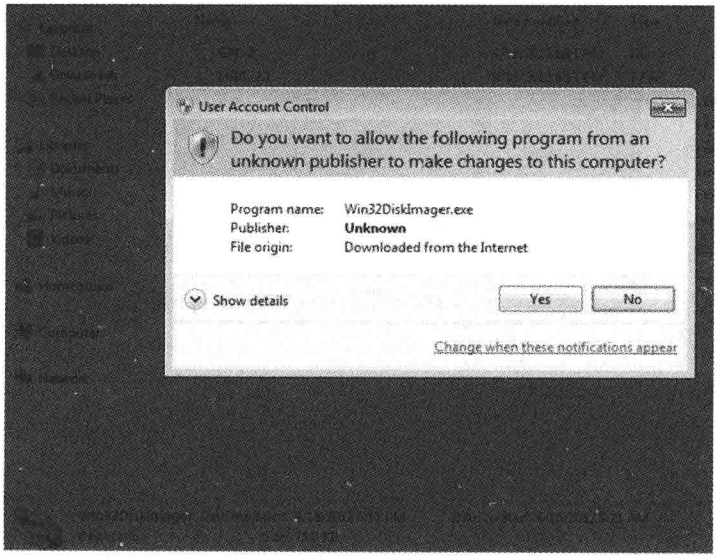


图 1-8　运行 Image Writer 提示的警告信息

4. 这是由于 Image Writer 需要申请一些系统权限来实现诸如磁盘擦写等操作，单击 Yes（是）继续。

5. 你有可能看到如图 1-9 所示的出错提示信息。别担心，在我们的测试中，这并不影响正常使用。单击 OK 继续。

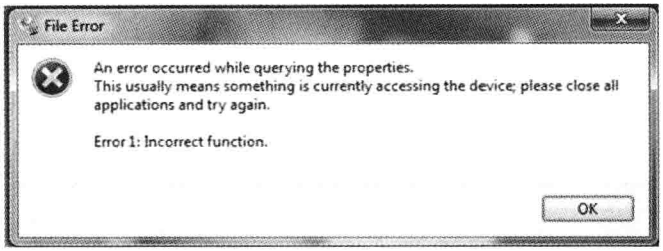


图 1-9　打开 Image Writer 可能会出现的出错提示信息

注意 Image Writer 只会在程序启动时尝试检测 SD 卡，所以我们必须在启动程序前将 SD 连接到电脑上。否则，即使你再插入 SD 卡，程序也无法检测到它。

6. 在 Device（驱动器）下拉菜单中选择 SD 卡相应的盘符（比如 G:\）。如果下拉菜单是空的，可以尝试关闭 Image Writer 后再重新打开。在选择盘符时请尽量谨慎，万一错选成了相机的 SD 卡，你数年积累的照片可能会毁于一旦。

7. 选择正确的驱动器（不确定的话，再打开“我的电脑”检查一次吧）和正确的磁盘镜像文件，单击 Write（写入）按钮。烧录需要几分钟时间，喝杯咖啡休息一下吧。图 1-10 是运行中的 Image Writer 截图，可以看到写入的是写作本书时最新的 Raspbian。

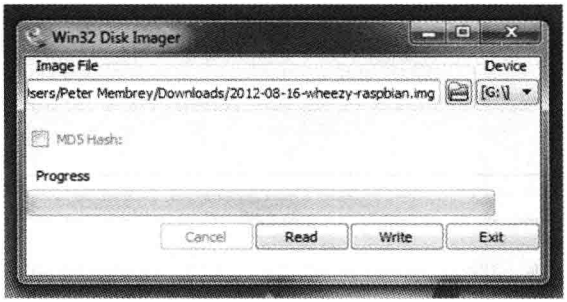


图 1-10　运行中的 Image Writer

8. 烧录完成后，将 SD 卡安装到树莓派上。

万事俱备了，快翻到“首次启动”部分开始折腾吧！

注意 细看图 1-10 可以发现，我们烧录的镜像文件后缀为.img。很多初学者忘记将镜像（.img）文件从 zip 压缩包中解压出来再烧录。如果再启动树莓派时再次遭遇黑屏，不妨再检查一下是否烧录了正确的文件。

在 Mac 系统上使用 dd 命令

我们很想为 Mac 用户介绍一种简单的方法，把 Raspbian 镜像写入到 SD 卡上，但很可惜，实在没法做到像在 Windows 系统上使用 Disk Imager 那么简单。Mac 系统上有 Disk Utility 工具，它很强大，却无法处理我们的 Raspbian 镜像文件。我们试图将镜像转换成.dmg 文件[1]，但还是不起作用。我们也尝试在网络上寻找一些工具来解决这个问题，但是最终我们发现，没有工具能明显地让这件事变得简单。

我们还是不得不使用命令行来解决问题。不过我们不准备在本章详细地解释将要执行的命令，后面的章节还会有介绍，所以先依葫芦画瓢吧。

打开终端（Terminal）

接下来的每个步骤，我们都会一边介绍，一边操作，最后会有一个完整的步骤列表。整个过程还是比较简单的，步骤也不是很多。

第一步，我们在应用程序（Applications）的实用工具（Utilities）目录中打开终端（Terminal）。打开终端后，我们需要使用 sudo 命令取得 root 用户权限，相当于 Windows 系统的 Administrator 用户。

由于直接操作底层设备是不安全的（比如，不能让病毒直接把数据写入硬盘），所以普通用户无法执行相关的命令。而我们需要将 Raspbian 镜像直接写入 SD 卡，必须能够直接操作硬件。root 用户对系统拥有完全的操作权限，所以我们可以使用 root 用户将系统镜像写入 SD 卡。通过下面的命令来提升为 root 用户。

```
sudo -i
```

这个命令让我们可以在终端中得到 root 用户权限的交互环境。你也可以通过 sudo（不带-i 参数）直接以 root 用户身份执行单条命令，但是这里我们需要执行多条，所以这样做会方便一些。按下回车键后，sudo 会询问密码（就是你登录时的账户密码），输入正确后，你就得到了 root 用户权限。

注意 得到 root 用户权限后，我们可以在系统上执行任意操作。但是请记住，这时除了可以将树莓派的系统镜像写入 SD 卡以外，我们也有可能会弄坏 Mac 系统。在 Unix 系统[2]中使用 root 用户时，你需要清楚自己在做什么，因为此时系统几乎没有任何保护。执行命令时请一定小心，仔细检查，确认没有错误再按回车键。而且在执行完命令后，请立即退出终端，以防万一。

使用终端写入镜像

OK，下面做什么？首先，我们遇到的第一个问题是确定 SD 卡对应的系统设备名称。为了避免一不小心抹掉主硬盘上的数据，我们需要先确定一下主硬盘的设备名称，防止误操作。

---

[1] 译者注：dmg 是 Mac 系统原生支持的镜像文件格式。

[2] 译者注：Mac OS X 也是一种 Unix 系统。

请在你的 Mac 上拔出 SD 卡，运行以下 mount 命令。

```
mbp:~ root# mount
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
mbp:~ root#
```

你应该能看到类似上面的输出，也有可能会有所不同。我们可以忽略 devfs 和 map 这几行，因为这些挂载项是操作系统的一部分。我们需要记住的是第一行，可以看到/dev/disk0s2 的挂载路径是/。这是系统盘，也就是主硬盘，一定注意不能随便写入这个设备。也许，你还会看到其他一些条目，那可能是因为你还挂载了网络驱动器或者移动硬盘。那你也应该把这些设备名称都记录下来，现在的挂载状态就是系统的基本状态，我们会和挂载了 SD 卡之后做对比。

接下来，在 Mac 上插入 SD 卡。你应该会在 Finder 边栏的设备（DEVICES）下看到刚刚插入的 SD 卡，如图 1-11 所示。

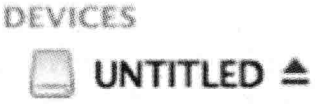


图 1-11　Finder 显示插入的 SD 卡

因为 Mac 会自动挂载插入的 SD 卡，所以我们可以直接回到终端再次执行 mount 命令，观察一下发生了什么变化。

```
mbp:~ root# mount
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
/dev/disk2s2 on /Volumes/UNTITLED (msdos, local, nodev, nosuid, noowners)
mbp:~ root#
```

我们发现多了一个挂载的设备，在上面的例子中显示的是 disk2s2，还能看到该设备的挂载路径是/Volumes/UNTITLED，这个设备正是接下来写入系统镜像的位置。

注意 BSD 系统将磁盘划分为多个分区（slice），这个例子里挂载的文件系统位于 disk2s2，也就是说文件系统位于第二个磁盘（disk2）的第二个分区（s2）。通过这个路径，可以让用户直接访问指定的分区。不过我们需要的是将数据直接写入设备，也就是磁盘，所以我们接下来将会对/dev/disk2 进行操作。

在进行下面的操作之前，我们要先解决一个小问题。由于 SD 卡的文件系统被自动挂载了，

我们无法直接向设备写入数据，而如果我们在 Finder 中卸载文件系统，SD 卡对应的设备也会被 Mac 系统推出，这样的话我们还是无法直接操作设备。所以我们需要通过 Mac 系统的 diskutil 命令手工卸载文件系统，并在设备中保留 SD 卡。Mac 系统也有 UNIX 标准的 umount 命令（这个命令在后面的章节会有详细解释），但是如果你想通过 umount 卸载 SD 卡，结果就是卸载失败并提示设备正忙。diskutil 可以避免这些问题，并成功地完成任务，所以我们通过执行下面的命令来卸载 SD 卡的文件系统。

```
mbp:~ root# diskutil unmount /dev/disk2s2
Volume UNTITLED on disk2s2 unmounted
mbp:~ root#
```

至此，我们在超级用户下得到了 SD 卡对应的设备名称，并从系统上卸载了 SD 卡的文件系统，这样我们就准备好向 SD 卡直接写入数据了。

我们将会使用一个古老的命令 dd，该命令用于从一个设备读入数据，并写入另一个指定的设备。该命令不会处理文件系统，也不会关心使用的设备是硬盘还是 SD 卡，而是直接在两个设备上读写二进制数据。我们会在最后一章详细介绍 dd 命令，这里我们只说明如何将系统镜像写入 SD 卡。

使用 dd 命令只需要知道两个参数，读取位置（if 参数）和写入位置（of 参数）。我们之前已经知道了写入位置是/dev/disk2，那么读取位置呢？如果你按照本章前面的步骤一步步做了，那么树莓派的系统镜像文件应该已经躺在下载（Downloads）文件夹中了，这就是我们的读取位置。首先，在终端中输入：

```
dd if=
```

之后在 Finder 中打开下载（Downloads）文件夹（或许你放在了其他文件夹中，请打开它），单击并拖动镜像文件到终端窗口。这样镜像文件的路径就会自动输入终端了，你应当会看到类似下面的显示。

```
dd if=/Users/myuser/Downloads/2012-08-16-wheezy-raspbian.img
```

这样我们就指定了读取位置，我们还需要为 dd 指定写入位置。在之前的操作中，我们已经得到了 SD 卡的设备名称：/dev/disk2。我们还要加上 bs=512 这个参数，让 dd 以较大的数据块写入 SD 卡（这样可以让写入过程快一些，否则可能需要等上 4 个小时）。最终整个命令是这样的：

```
dd if=/Users/miggyx/Downloads/2012-08-16-wheezy-raspbian.img of=/dev/disk2 bs=512
```

按回车键执行命令，dd 需要一些时间来完成整个操作（可能还是需要 1、2 个小时）。由于系统镜像有 2GB 大，而 SD 卡的写入速度并不是很快，所以我们的 MacBook Pro 也花了接近 2 个小时才完成整个镜像的写入。你可以出去逛一圈，回来之后应该能看到 dd 完成了写入，终端回到了命令提示符状态。

在 Mac 上写入镜像确实比在 Windows 系统上麻烦许多，不过仔细观察一下这些命令，你会发现通过命令执行操作更简单，更准确，第 3 章我们还会详细介绍命令行的其他优点。

我们将前面所有的操作步骤整理了一下。如果对于之前的介绍还有什么问题，或者你还有什么不太明白的，可以参考下面的内容。

```
Last login: Wed Sep 19 13:48:40 on ttys000
mbp:~ pmembrey$ sudo -i
Password:
mbp:~ root# mount
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)

<<SD Card inserted>>
mbp:~ root# mount
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
/dev/disk2s1 on /Volumes/UNTITLED (msdos, local, nodev, nosuid, noowners)
mbp:~ root# diskutil unmount /dev/disk2s1
Volume UNTITLED on disk2s1 unmounted
mbp:~ root# dd if=/Users/pmembrey/Downloads/2012-08-16-wheezy-raspbian.img of=/dev/disk2
bs=512
mbp:~ root#
```

到这里，我们准备好了 SD 卡，让我们来准备启动树莓派吧。

1.5　首次启动

前一节的操作还是比较复杂的，希望你一次就能成功。不过应该不算太困难，毕竟整个过程没有多少需要调整的参数。

完成 SD 卡写入操作后，就可以准备启动我们的树莓派了。连接好键盘、鼠标和显示器，把写入了系统镜像的 SD 卡插入树莓派，最后接好电源，我们就可以启动了。如果一切正常，你应该看到了如图 1-12 所示的画面。

如果你的确看到了如图所示的画面，那么恭喜你，你的树莓派已经启动了。如果没有的话，那可能是有些地方出错了。

首先你可以检查一下树莓派上的红色 LED 灯是否亮着。红灯看似很明显，但有时却是最不被关注的。如果红灯没有亮起，你需要检查一下是否接错了电源线，或者电源没有打开。可以仔细检查一下电源的问题，确定没有问题之后再去向别人提问。

如果电源没有问题，第二项你可以检查的是树莓派是否正确地连接了显示器或电视。如果使用电视，需要注意大多数电视都有多种输入方式可选，而如果使用显示器，需要注意显示器一般都没有 HDMI 接口。这里很难给出一个明确的步骤来检查显示器的连接情况，不过你可以尝试在显示器的多个输入方式之间来回切换，也许就能看到树莓派的启动画面。比如我们在尝试使用电视连接树莓派时，虽然电视上写着 HDMI 接口 2，但是最终我们却在 HDMI 接口 3 上看到树莓派的画面。所以试试看，也许仅仅是很小的错误。



图 1-12 树莓派首次启动

最后，如果前两项都没有发现问题，可以检查一下 SD 卡是否完全插入了卡槽。如果没有正确地插入 SD 卡，树莓派是无法启动的，你能看到的只是一个黑黑的屏幕。如果你确定 SD 卡也完全插入了，却还是无法正常启动，那有可能是在写入系统镜像时出了问题。你可以重复一次上一节的步骤。如果最后各种尝试都无法正常启动你的树莓派，那我们也无能为力了。不过没关系，树莓派的论坛上还有很多人愿意帮助你，可以访问 www.raspberry.org，进入论坛，在 Basic Usage（使用基础）板块或 Using the Raspberry Pi（使用树莓派）的 Setup（安装）板块中寻找答案。

1.6 配置树莓派

如果你已经成功地启动了树莓派，那你应该会看到如图 1-13 所示的文字版配置界面。在这里，你可以为首次使用树莓派进行一些配置，配置完成后，再次启动树莓派就会直接进入桌面了。

我们将会介绍一些基本配置选项，这样你就可以尽快地准备好树莓派。首先，你可以调整一下启动选项，让你的树莓派下次启动时直接进入桌面（在 boot_behavior 选项中选择 desktop）。

扩展文件系统

第一个介绍的选项是 expand_rootfs，该选项可以将整个 SD 卡空间用于文件系统。我们下载并写入 SD 卡的系统镜像大小大概是 2GB，那么无论你的 SD 卡实际是 8GB 或者更大，第一次启动时，你在树莓派中看到的文件系统大小也差不多这么大。通过启用该选项，我们就可以完全利用 SD 卡的空间。通过这个功能，树莓派系统无需为不同大小的 SD 卡做修改，统一的系统镜像即可适用于各种情况。你可以通过键盘上下选择，高亮 expand_rootfs 选项后按下回车键。之后，会弹出一个窗口告诉你，分区大小已经被修改，在下次启动时扩展文件系统大小将会生效。再次

按回车键回到配置菜单。

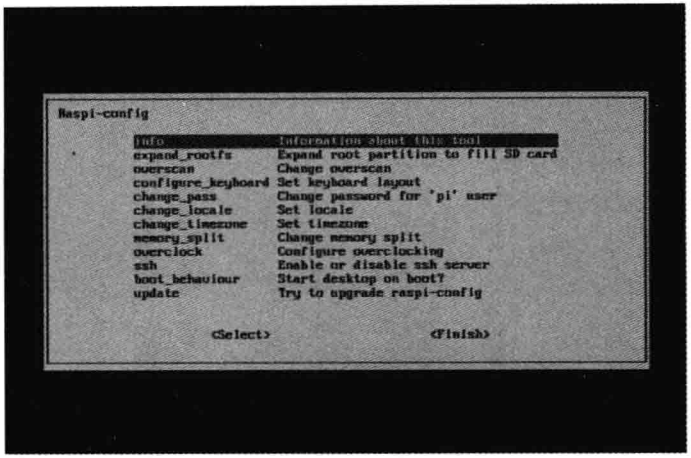


图 1-13　首次启动的配置界面

配置键盘

下一个需要说明的选项是 configure_keyboard，默认情况下，树莓派使用的是 UK 键盘映射[1]。如果你使用的是其他语言的键盘，那么可以通过这个选项调整键盘映射。不过该选项的第一屏——选择键盘类型，就是一个复杂的选择，树莓派提供了数十种键盘类型，但是你真地知道你的键盘是 104 键还是 105 键？我们选择了默认的 Generic 105 key (Intl) PC 键盘类型，如图 1-14 所示。

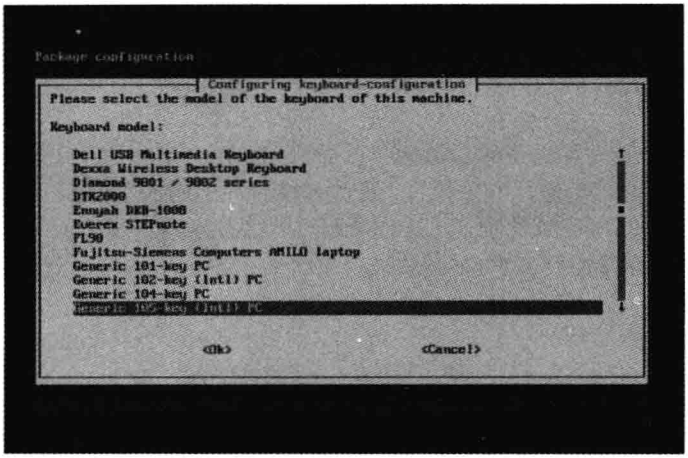


图 1-14　选择键盘类型

[1] 译者注：之所以默认为 UK 键盘，是因为树莓派来自英国。

如果你能在这个列表中找到你使用的键盘，那你应该选择正确的类型。不过，你要是不确定或者不清楚的话，默认的选项也没有问题。选好键盘类型后，按回车键进入下一屏幕。

下一屏是键盘映射的选择。默认的选项是 UK 键盘映射（见图 1-15），如果你使用的不是 UK 键盘，可以通过键盘选择 Other（其他），按回车键，之后会有一个列表让你选择合适的键盘映射，按回车键确认后，进入下一屏幕。

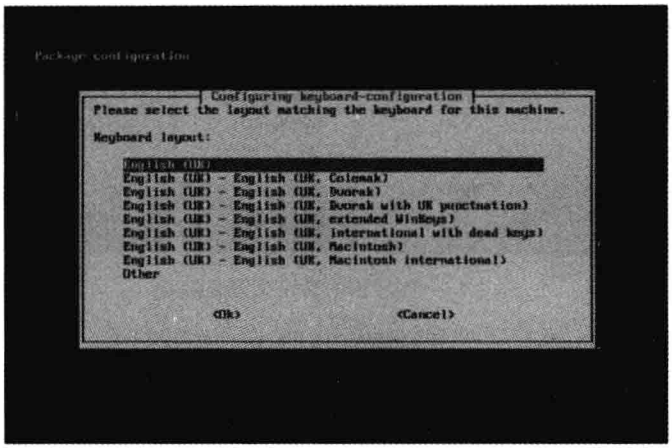


图 1-15　选择键盘映射

完成键盘映射选择之后，我们见到了第 3 个选择列表，估计你对这些选项会感到很迷茫（见图 1-16）。这个列表描述的是键盘特殊键的类型，我们写作本书的时候也觉得很复杂。所以在这里使用默认选项就好了，除非你确实知道你的键盘对应的是哪个选项。对于大多数人来说，直接按回车键，确认默认选项即可。之后进入了下一屏幕的列表（见图 1-17），与刚才描述的类似，我们就使用默认选项，按下回车键。

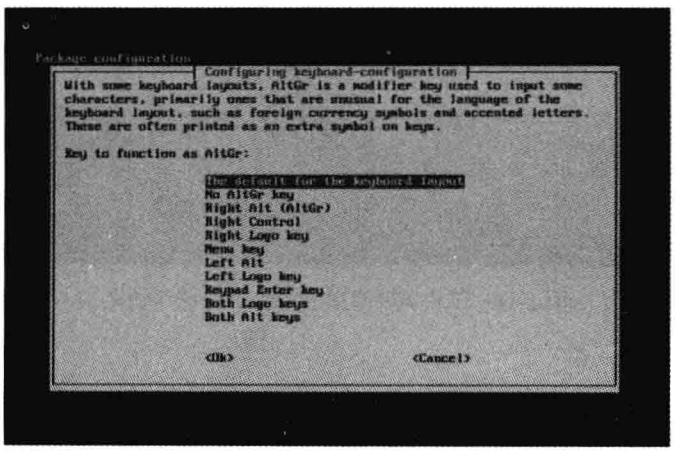


图 1-16　自定义键盘选项

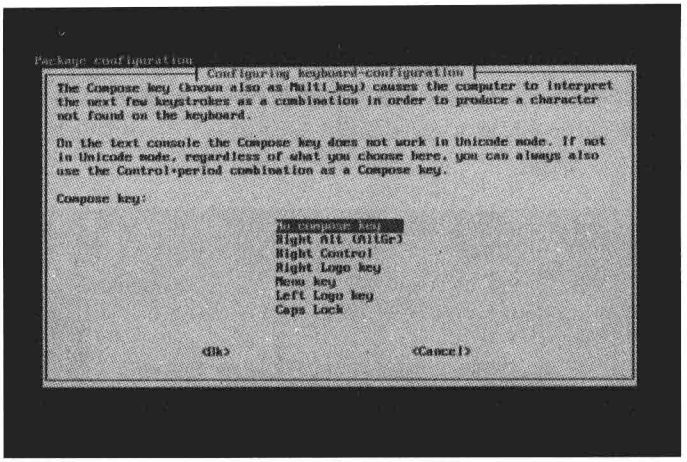


图 1-17　更复杂的自定义键盘选项

下一屏幕（见图 1-18）对于早期的 Linux 用户来说会比较熟悉。以前使用 Linux 图形界面（GUI）时，如果图形界面崩溃，你可以通过 Ctrl+Alt+Backspace 组合键来迅速重启图形界面。随着图形界面越来越稳定，Linux 的使用越来越普遍，这个快速重启图形界面的组合键已经逐渐停止了使用。所以这里默认的选项是 No，你可以确认这个默认选项，按回车键即可。如果需要，你可以之后再次进入配置界面打开这个选项。

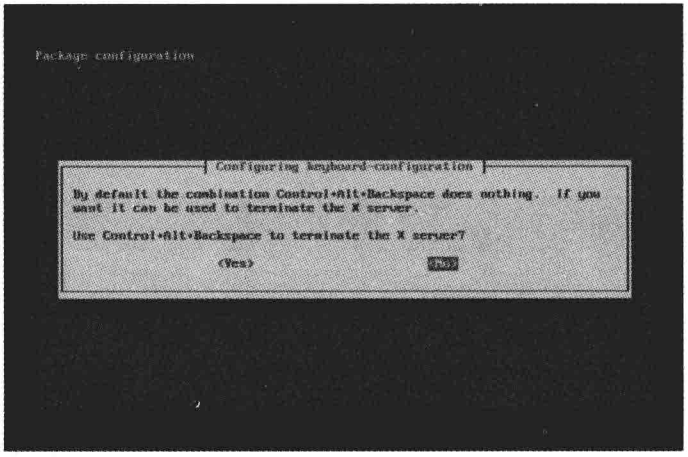


图 1-18　是否需要启用关闭 X 服务的组合键

修改密码

下面让我们来修改 pi 用户的密码，在本书中我们将会一直使用这个账户名，而且对于大多数树莓派的用户来说，这个账户已经足以满足需要了。每一个树莓派的 pi 账户的默认密码都是

raspberry，所以任何人都可能登录你的树莓派。虽然对于树莓派这种实验平台，安全性似乎不太重要，但最好还是修改一下默认密码。在后面的章节，我们还会指导你将树莓派连接入互联网，如果你的树莓派没有修改密码，就很容易被别人入侵。所以还是在这里先修改一下密码，通过键盘高亮 change_pass，按下回车键，之后你应该能看到如图 1-19 所示的界面。

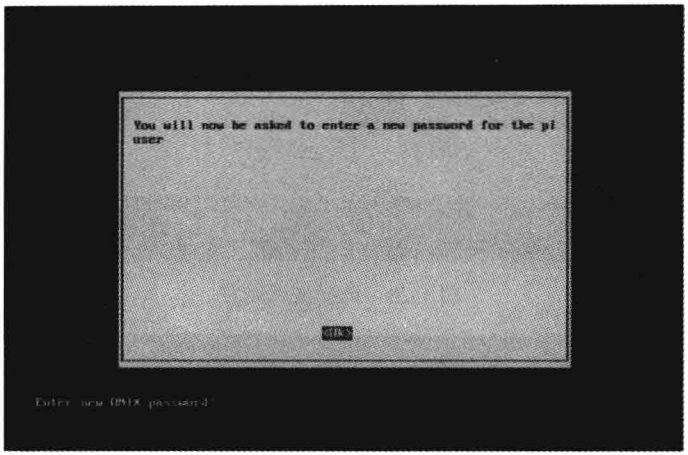


图 1-19 修改密码

因为这个密码修改工具调用的就是 passwd 命令，所以你看到的是这种简单的文字界面，而不是漂亮的图形化界面。在这里输入新密码（密码是大小写敏感的，*HeLLo*、*hello* 和 *Hello* 是不同的），之后按回车键确认。需要注意，输入密码时屏幕上不会有任何提示，当你按下回车键确认后，会弹出窗口让你再次输入新密码作为确认，完毕后将回到配置菜单。

配置区域设置

接下来，我们调整区域设置。设置区域可以确保 Linux 系统以正确的方式显示数字分隔符和货币符号。比如在美国，数字习惯上写成 1,000.00 的形式，但是在欧洲的一些地方，数字会被写成 1.000,00。通过区域设置，你可以让系统符合你的特殊要求。使用键盘高亮 change_locale 选项，按回车键。

之后你应该会看到一个区域名称列表（见图 1-20）。区域名称一般都由两部分组成，前两个字母表示的是该区域使用的语言，下画线风格的后一半内容表示的是语言的不同方言。例如，en_US 表示使用美式英语，en_GB 则表示使用英式英语。你可以选择最接近所在地区使用的语言和习惯组合，通过键盘高亮你需要的组合，按空格键添加选择。也许你会选择同一区域的 ISO-8859 和 UTF-8 选项，不过需要注意的是，你选择添加的区域设置越多，树莓派就需要越多时间来准备和生成这些区域设置。当你确定好区域设置之后，通过 Tab 键高亮 OK，并按下回车键进入下一屏幕。这里会让你选择默认区域设置（见图 1-21），选择好后，按回车键确认，之后将回到配置菜单。

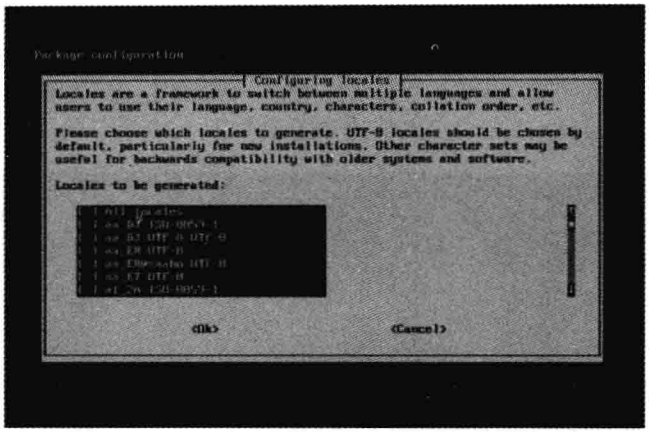


图 1-20　选择区域设置

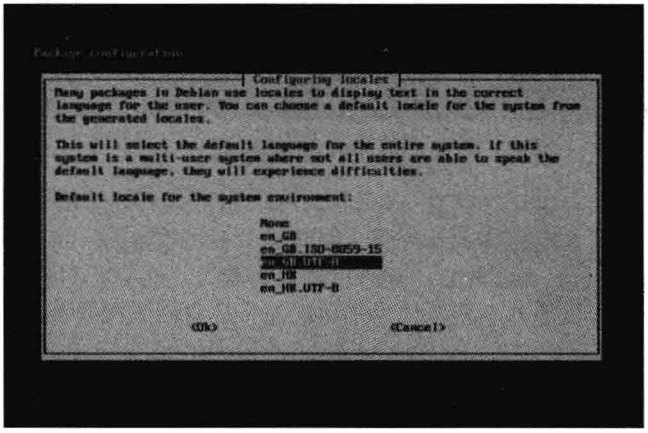


图 1-21　选择默认区域设置

调整时区

这节我们介绍 change_timezone 选项，在配置菜单中选择该选项，将会进入地区列表（见图 1-22）。这里，我们配置的是作者 Peter 的树莓派，我们选择亚洲。确认后，下一屏幕将让你选择你将使用的时区（以城市名称为代表，如图 1-23 所示），选择最接近你所在时区的城市[1]，按回车键。

分配内存

对于 memory_split 选项，我们需要解释一下。树莓派有两个处理器：CPU（中央处理器）和

[1] 译者注：比如中国可以选择 Shanghai 或者 Chongqing。

GPU（图形处理单元）。CPU 负责执行通用计算，GPU 负责执行显示渲染、2D 和 3D 图形加速以及高清视频解码。GPU 对图形的高速处理可以减轻 CPU 的计算负担，所以树莓派安装 GPU 可以保证性能，并降低成本。

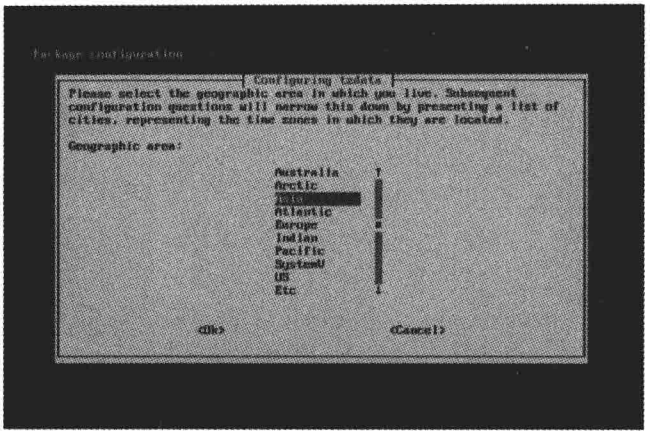


图 1-22　选择地区

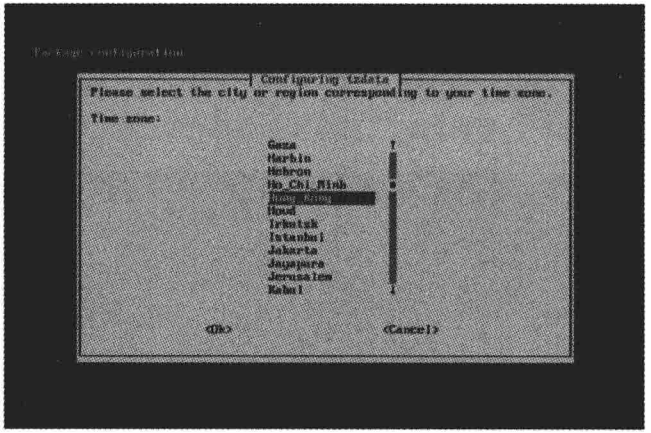


图 1-23　选择时区代表的城市

那么我们为什么要分配内存呢？因为 GPU 作为独立的计算单元也需要内存，而树莓派只有 256MB 内存，所以 CPU 和 GPU 需要共享这些内存。默认情况下，树莓派将 192MB 内存划分给 CPU 使用，64MB 划分给 GPU。不过，如果你将树莓派用作服务器，偶尔连接一下显示器的话，那就没必要分配这么多内存给 GPU 了。

另一种情况下，如果你将树莓派当作迷你桌面电脑，而且还要播放高清电影的话，那么你应该再多分配一些内存给 GPU（比如 CPU 和 GPU 各分配 128MB）。由于不同的使用场景，树莓派很难为你决定应该如何合理地分配内存，所以它提供了这个选项，让你来自行调整。

高亮 memory_split 选项，按回车键进入内存分配界面，如图 1-24 所示。对于“入门级”的

玩家，我们推荐使用默认的 192MB/64MB 分配方案。如果你需要自行调整，也可以在这里输入其他数组，完成后按回车键确认。

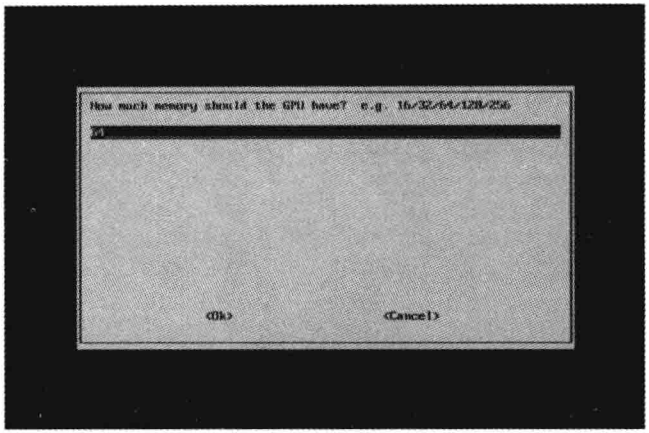


图 1-24　分配内存

配置完成！

喔哦，到这里，我们终于完成了树莓派的初始配置。你现在可以重启一下树莓派，并准备开始使用吧。在配置菜单中通过 Tab 键高亮 Finish，并按下回车键，树莓派将会自动重启。如果你之前调整了扩展文件系统选项，那启动时应当能看到如图 1-25 所示的显示内容。

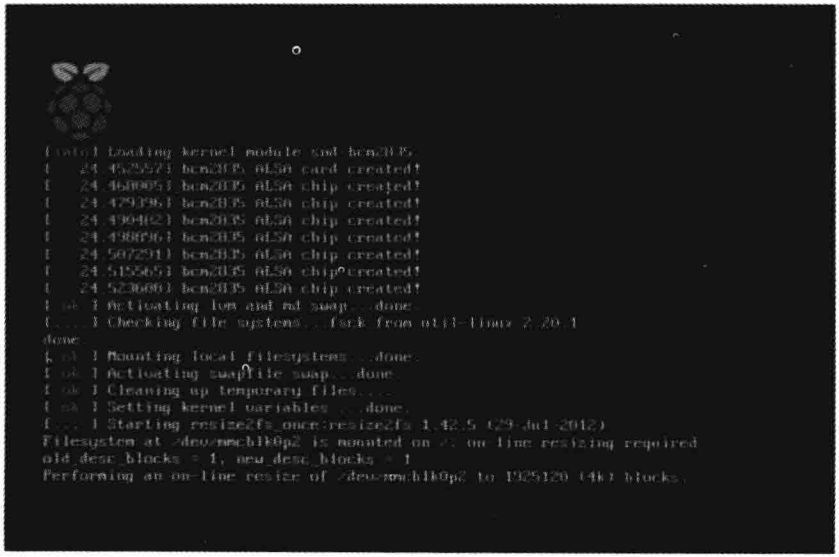


图 1-25　启动时显示扩展文件系统

扩展文件系统的过程受你所使用的 SD 卡大小影响，由于调整过程没有什么屏幕提示，你会觉得这次启动非常缓慢，不过一旦调整完毕，下次就会恢复正常的启动速度。

经过这么多设置和等待，当树莓派启动完毕，你应该会在显示器上看到如图 1-26 所示的图形界面。

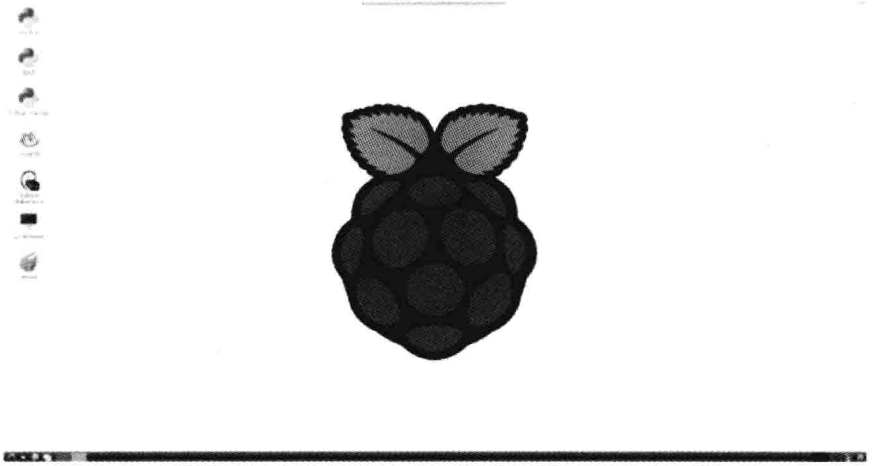


图 1-26　树莓派的图形界面

恭喜你，你已经成功地启动了树莓派！如果你没有能够进入图形化的桌面，你可能需要再次调整之前的配置，在字符登录界面使用用户名 pi 和之前设置的密码登录树莓派，输入以下命令再次进入配置菜单（比如，在配置菜单中再次调整 boot_behavior 选项）：

```
$ sudo raspi-config
```

配置完毕后，可以通过以下命令尝试启动图形化界面：

```
$ startx
```

本章小结

本章我们首先介绍了如何从零开始将树莓派和外设连接好并最终让其运行起来，我们详细地讲解了其中的细节，有人觉得这些都很简单，但我们往往会在这些简单的步骤上犯错。我们还列出了树莓派自带了什么组件，需要你自己安装什么组件，了解了这些你能更好地把玩树莓派。

之后我们简要地介绍了 Linux 系统，包括什么是发行版，为什么本书将只介绍 Raspbian。我们还讲解了如何下载 Raspbian 并将其写入 SD 卡。

最后完成 SD 卡写入后，我们启动树莓派，介绍了如何在首次启动时设置树莓派，最后我们进入了桌面，看到了带有树莓派的 Logo 背景。

下一章，我们将带你探索树莓派的桌面系统，介绍 Raspbian 自带了哪些应用程序，还会一起学习如何自定义树莓派。

第 2 章

开始探索

读完本章后，你将了解 Linux 系统，学会使用命令行环境。但在开始时，我们还是要介绍一下树莓派的图形界面。虽然树莓派的图形界面没有 Windows 或者 Mac 那么绚丽，甚至还不如一些 Linux 发行版，但是它轻巧、快速、易用，能够完成各种常用的操作，最重要的是它能够运行在这么小巧的计算机上。

本章中，我们将会带你浏览一下桌面，了解一下系统中安装的应用程序。有些应用比较常用，有些用得很少（比如你可能很少会用到 Python 开发环境）。我们还会看看菜单中还隐藏了什么有用的功能。

2.1 欢迎使用 LXDE

Linux 操作系统和 Windows 系统重要的不同点在于它并不集成某个特定的图形界面，而是将图形界面作为一个独立的应用程序（客户端/服务器架构）提供给你。这个应用程序就是著名的 X-server，它还是最难以配置的 Linux 组件。几年前，想在 X-server 中正常使用鼠标滚轮，需要费很大的功夫才能做到。不过现在，X-server 已经改进了很多，基本不再需要什么设置就可以开始使用了。这里我们不会深入地介绍它，不过如果你很感兴趣的话，可以访问 http://en.wikipedia.org/wiki/X_Window_System[1]进一步了解。

X-server 为图形界面提供了一个开发运行框架，但是它并不能为你显示出一个漂亮的桌面。图形界面是由一个桌面环境套件提供的，可以自由地在多种桌面环境中选择，提供多种选择也是 Linux 系统的惯例。对于大多数用户来说，选择使用某个 Linux 发行版就选择好了图形桌面环境，因为许多发行版都会为你安装一个默认的桌面环境。一些最新版的桌面环境提供了各种高级的功能和绚丽的特效，这会消耗很多系统资源。虽然对于主流的桌面计算机和笔记本来说不算什么，但是要在树莓派上运行就比较困难。

其实许多旧电脑也很难运行最新的图形界面，所以这不只是树莓派会遇到的问题。那么除了扔掉那台古老的电脑以外，还有什么选择呢？你可以选择安装一个轻量级的图形桌面，许多低功耗电脑设计时就考虑到了尽量省电，包括低功耗的图形界面，所以可以在树莓派上运行这些设计

[1] 介绍 X Window 的维基百科中文页面：http://zh.wikipedia.org/wiki/X_Window 系统。

简洁的应用，代替那些复杂绚丽的图形界面。

想要控制计算机的成本时，最好的方法就是放弃那些绚丽的图形界面，你真地需要透明窗口和 3D 时钟么？像 LXDE 这种简洁的桌面就是很好的选择。LXDE（Lightweight X11 Desktop Environment 的缩写）是专为低功耗计算机和低端硬件而设计的，它尽量降低对 CPU 的使用，节省更多的内存，关闭各种不必要的特效，并提供所有必需的功能。

LXDE 是很容易学的图形界面，如果你准备在树莓派上主要使用图形界面的话，LXDE 是很好的选择。所以接下来，让我们来一起了解下 LXDE。

LXDE 里有些什么

安装并初始设置好树莓派后，启动它，你就会看到如图 2-1 所示的桌面。

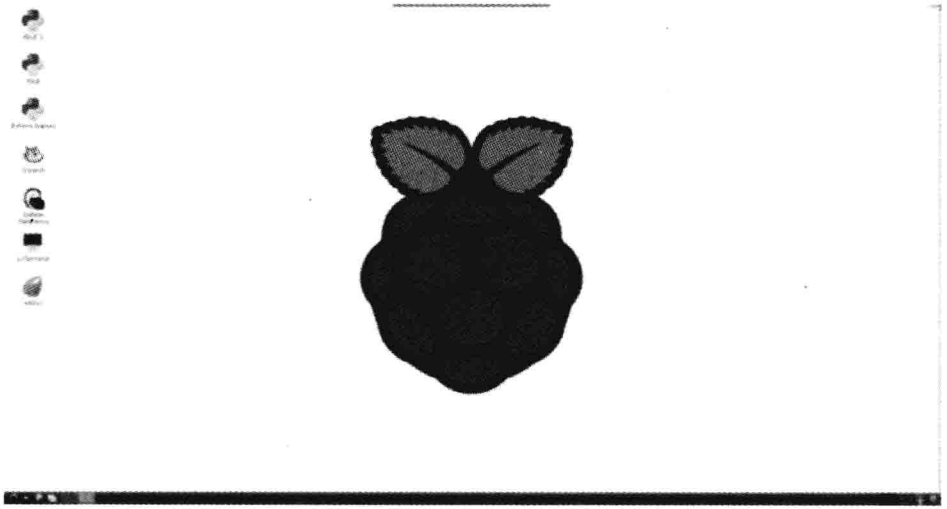


图 2-1 首次启动树莓派

桌面上图标的排列可能和你的树莓派略有不同，树莓派也在不断地改进和升级，所以你拿到的树莓派可能还会有一些新图标。

有些细节在这样一张桌面截图上很难看清楚，所以我们会用局部细节的图片来介绍，我们会使用一些你熟悉的词汇来描述图形界面的元素。下面就让我们来看看开始按钮、菜单和任务栏，也就是屏幕的左下角。

开始按钮和任务栏的左下角

左下角的第一个按钮（见图 2-2）图标是 LXDE 的 Logo，它的功能相当于 Windows 中的开始按钮。第二个按钮可以启动 PCManFM——LXDE 的文件管理器，可以通过它整理文件。第三个按钮（地球图标）可以启动浏览器，树莓派默认安装的浏览器是 Midori。和其他的应用一样，你会发现它们都是以轻量级为设计目标的。



图 2-2　开始按钮

这些应用大多和你之前用过的应用很类似，不过接下来要介绍两个比较特别的功能，它们不是单独的应用，而是用于桌面环境管理的工具。首先介绍的是全部最小化（图标是两个叠在一起的窗口），单击后所有当前打开的窗口都会最小化，不过需要注意的是，再次单击并不会恢复之前窗口打开的状态。这个按钮还有第二个功能，通过鼠标中键单击它（按一下滚轮），所有的窗口都会收起，收起的窗口都缩小到只有标题栏。

下一项是桌面选择区，对于微软的 Windows，默认情况下只有一个桌面，而且大多数用户都不知道其实还可以打开更多桌面。Linux 系统则是默认提供了多个桌面，而且几乎所有基于 UNIX 的操作系统也都拥有这个功能。

使用桌面选择区，可以在多个桌面之间切换，每个桌面都可以独立地打开应用。比如可以将所有的工作应用放在一个桌面中，同时将聊天应用和浏览器放在另一个桌面中。LXDE 默认提供 2 个桌面，也可以根据需要，打开更多桌面。

任务栏其余区域将用于显示启动的应用，这很像 Windows 系统，所有运行中的应用（有图形界面的应用）都会显示在任务栏中，可以通过任务栏进行切换。

任务栏的右下角

如图 2-3 所示，任务栏的右边还有一些图标。第一个看起来像是个黑色方块，这其实是 CPU 监控器，当树莓派运行负荷增大时，这个图标就会变绿，它可以实时提醒你树莓派的 CPU 使用率。你可以尝试快速地移动鼠标，移动得越快，CPU 使用率就越高，这个图标就会变得越绿。如果树莓派运行变慢，你可以注意一下，看看是不是超出了 CPU 的处理能力。



图 2-3　任务栏的右下角

CPU 监控器旁边是时钟，之后是锁屏按钮，该图标看起来像是显示器中有一个锁。锁屏按钮可以防止你朋友趁你离开时通过树莓派在你的微博上乱发状态。不过要让锁屏按钮真正发挥作用，最好修改一下默认密码。

最右边是电源按钮，如图 2-4 所示，可以通过这个按钮重启或者关闭树莓派，如果是通过命令行登录树莓派，并使用 startx 手动启动 LXDE 的话，那看到的则是 Logout（注销）和 Cancel（取消）。

Shutdown（关机）或者 Reboot（重启）可以迅速地关闭树莓派系统。不过需要注意的是树莓派运行的是 Linux 系统，Linux 是一种多用户操作系统，也许同时还有其他用户在使用树莓派，比如在后台播放视频，如果这时关闭或者重启了树莓派，可能会影响到别人的使用。



图 2-4 单击电源按钮

目前为止，屏幕上能看到的都介绍过了。让我们继续深入下去，下一节我们将会介绍开始菜单。

2.2 开始菜单

自从 Windows 95 引入开始菜单之后，各种操作系统都开始引入类似的功能（唯独 Mac 系统没有）。有人对所谓微软发明了开始菜单这种说法嗤之以鼻，并认为 Linux 系统中采用的类似菜单和 Windows 的开始菜单毫无关系。我们认为好用才是硬道理，开始菜单的概念得以普及并非由于谁发明了它，而是它的确很实用。如图 2-5 所示，这就是 LXDE 的开始菜单。

开始菜单简洁明了，列表中展示了所有已安装的应用程序，并按类型进行了分组，包括 Accessories（附件）、Education（教育）、Internet（网络）、Programming（编程）和 System Tools（系统工具）。除了 System Tools，其他分组没什么特别需要介绍的，所以让我们先看看开始菜单的其他部分。

应用程序列表之下是 Preferences（设置）和 Logout（注销），顾名思义设置可以对系统进行一些调整，注销则退出当前的用户。开始菜单中还剩下一个按钮 Run（运行）。单击运行，会弹出一个对话框，如图 2-6 所示，这里可以执行命令或者启动应用，它的用法和 Windows 系统上的“运行”很像，而且还有自动完成功能。

这里可以只输入字母 gno，自动完成功能会匹配已知的命令，例如 gnome-text-editor。命令名称经常是很长的（比如例子中的），自动完成是个很实用的辅助功能。

附件

接下来浏览 Accessories（附件）中有哪些应用程序。安装了一些应用程序后，这个分组中的

应用程序也可能会增加，而且新版的树莓派显示的可能会略有不同，在我们的树莓派中，Accessories 菜单如图 2-7 所示。

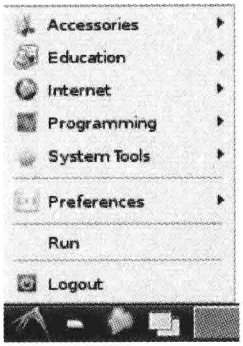


图 2-5　开始菜单

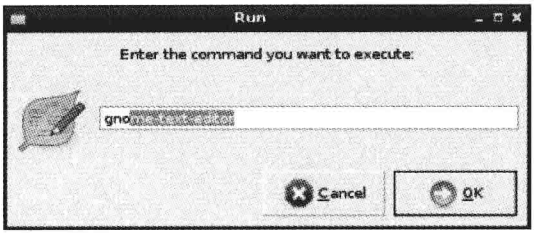


图 2-6　运行对话框和自动完成功能

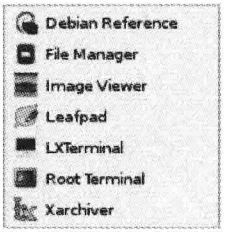


图 2-7　附件菜单

Debian 参考手册

Accessories 菜单的第一项是 Debian Reference（Debian 参考手册），树莓派的 Raspbian 系统是基于 Debian 开发的，该手册可以给你一些使用指导。这个手册由一系列 HTML 文档组成，所以单击图标后，会自动打开 Midori 浏览器，单击如图 2-8 所示页面中的第一个链接。

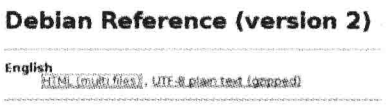


图 2-8　选择打开左边的 HTML 版本手册

图中的第二个链接指向的是该手册纯文字版本，这个版本阅读起来显然没有 HTML 版方便。单击了第一个链接后，应该会看到如图 2-9 所示的页面。

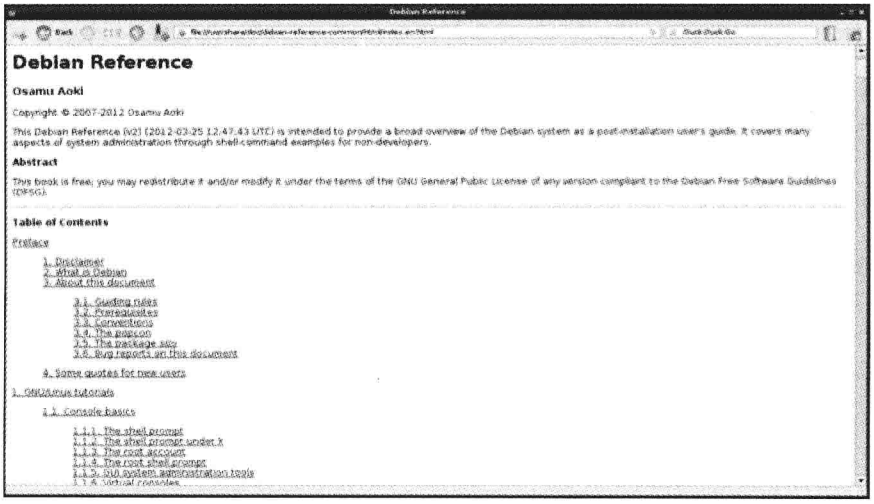


图 2-9　Debian Reference

可以像浏览网站一样阅读这个手册，由于是离线的版本，所以搜索起来不是很容易，不过通过目录，也可以很快地找到你所需要的指导。

文件管理器

LXDE 集成了 PCManFM 作为文件管理器，除了通过 Accessories 菜单，你也可以使用任务栏图标启动它。PCManFM 是一种轻量级文件管理器，不过麻雀虽小五脏俱全，功能和微软 Windows 资源管理器类似，如果你之前一直使用 Windows 操作系统，那么你对 PCManFM 的界面不会感到很陌生（见图 2-10）。

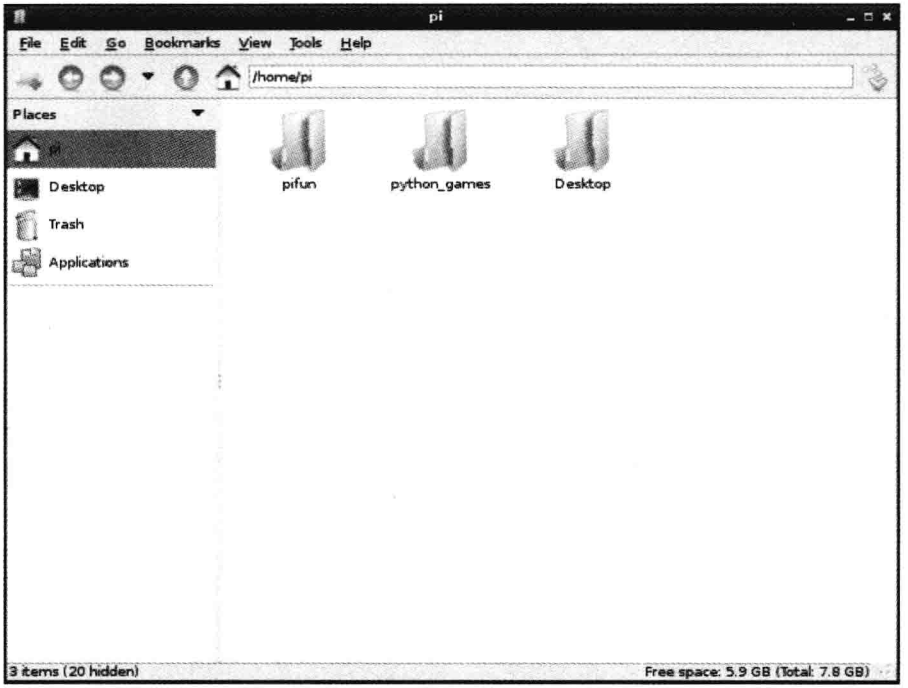


图 2-10 PCManFM 文件管理器

打开 PCManFM 文件管理器之后，它会默认打开主目录（/home/pi），状态栏右下角显示磁盘的使用情况，左边侧栏显示一些常用的位置链接。可以双击打开目录，使用工具栏的图标在目录间导航。第 4 章会详细介绍如何管理文件。

图像查看器

Image Viewer（图像查看器）是树莓派默认的图片浏览工具，双击打开图片时会自动启动，Image Viewer 提供了图片浏览的基本功能，界面如图 2-11 所示。



图 2-11　图像查看器的界面

Image Viewer 工具栏上有许多功能按钮，最左边的两个箭头按钮用来在相册中切换前一张和后一张图片，旁边的播放按钮可以启动幻灯片浏览。

第二块区域是一组浏览控制按钮，包括：

- 放大、缩小。
- 适合屏幕。
- 按原尺寸显示（如果是一张很大的图片，例如数码相机拍摄的照片，可能需要滚动浏览）。
- 全屏幕浏览。

第三块区域是一组图片调整按钮，前两个按钮的功能是旋转图片（分别为向左旋转和向右旋转），后两个按钮的功能是水平翻转和垂直翻转图片，比如用手机前置摄像头拍摄的照片效果是镜像的，可以通过水平翻转将其调整正确。

最右边的区域是文件控制按钮，包括打开、保存、另存为和删除按钮。最后两个按钮分别是打开设置面板和关闭图像查看器。

Leafpad

Leafpad 是一个文本编辑器，它也体现了 Raspbian 的简洁设计风格。从外观来看，Leafpad

很像微软的记事本，如图 2-12 所示。

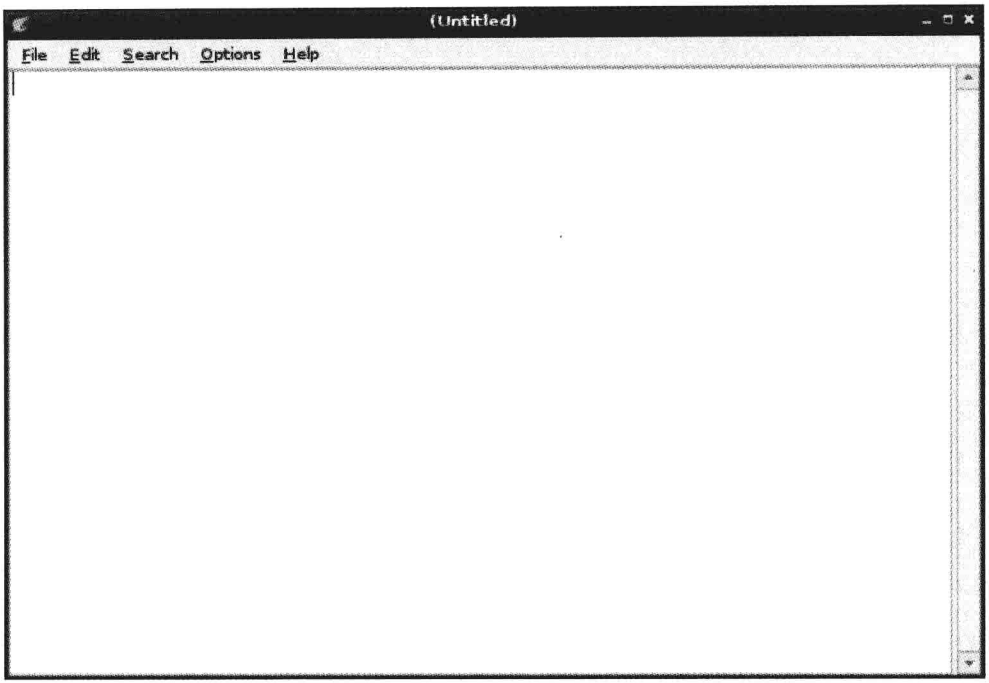


图 2-12 Leafpad 界面，很像微软的记事本

Leafpad 虽然设计简单，不过仍然提供了完善的功能，包括搜索和替换、自动换行等，基本满足了简单文本编辑的需要。

注意 如果觉得 Leafpad 缺乏必需的功能，可以直接阅读第 6 章，我们将会介绍更加强大的命令行文本编辑器。

LXTerminal

LXTerminal 是一种在 Linux 上运行的终端模拟器，可以使用它直接执行各种命令，第一次使用命令行看起来有点复杂。不过我们会在下一章详细介绍命令行的使用方法，所以不用着急。

LXTerminal 的界面也很简单（见图 2-13），不过说实话，终端模拟器还需要什么华丽的界面么，有闪烁的光标就足够了。

需要注意的是，绿色字体并且以美元符（$）结尾的命令提示符表示当前是普通用户模式，拥有较高的安全性（你可以随意执行命令，几乎不会损坏操作系统）。命令提示符的另一种模式我们接下来就会介绍。

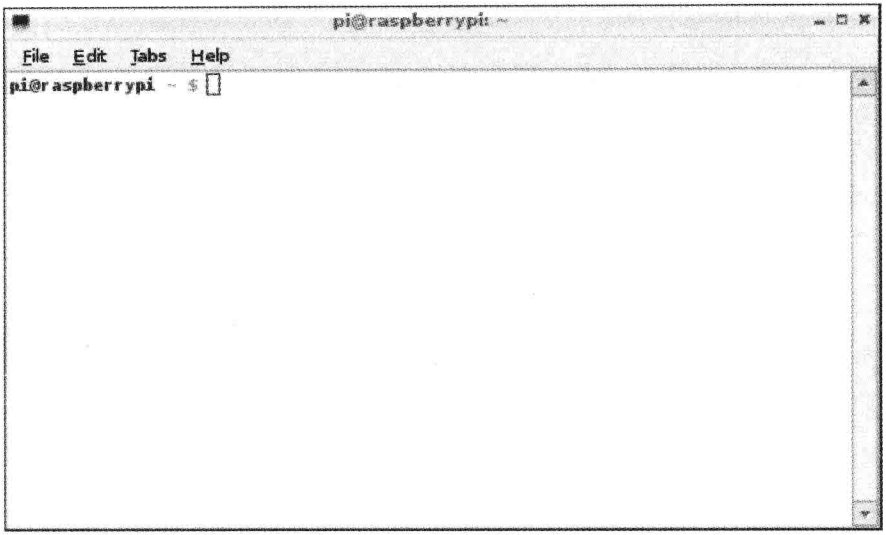


图 2-13　LXTerminal 的界面

Root 终端

Root Terminal（Root 终端）与 LXTerminal 的使用方法是一样的，所不同的是 LXTerminal 以普通用户身份执行命令，而 Root 终端则是以系统管理员身份（root user，根用户）执行命令。首次启动 Root 终端时会看到如图 2-14 所示的提示信息。

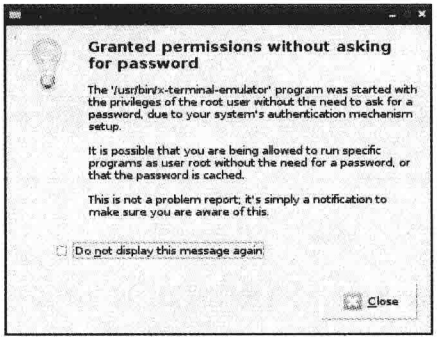


图 2-14　Root 终端首次运行提示信息

这个提示看起来像是个错误提示，其实它只是告诉你系统将允许应用在不输入密码的情况下执行高权限命令（此处是允许 LXTerminal 获得 root 权限）。需要注意的是，虽然在树莓派上使用这样的高权限会带来很大的便利，但是如果计算机中存放了敏感数据，并且他人很容易访问，保留密码保护才是明智之举。

Root 终端和 LXTerminal 在很多方面都十分相似，但它们的命令提示符有些不同（见图 2-15）。

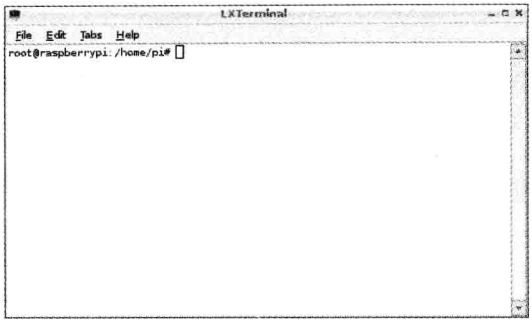


图 2-15 以 Root 权限运行的 LXTerminal

可以看到命令提示符是以井号（#）而不是美元符（$）结尾的，这意味着你可以执行高权限操作，这种权限下我们可以轻而易举地把系统折腾得分崩离析。因为以 root 用户身份执行操作时，即便你想删除硬盘上的所有文件，Linux 也会忠实地执行。所以“权利越大，责任越大”，除了少数必须的情况，请谨慎使用 root 权限。在本书中会多次提及这一话题。

XArchiver

Image Viewer 这类应用的功能一目了然，而像 XArchiver 则有多种不同的功能。可以用它压缩或解压多种格式的文件，这在处理网络下载文件时十分有用。图 2-16 所示是 XArchiver 的截图。

可以通过双击一个压缩包启动 XArchiver，目前我们只需用到从右边起第二个按钮：Extract all（全部解压）。

教学

Education（教学）菜单只有两个条目，但这两个应用都十分有趣，值得一试。菜单如图 2-17 所示。

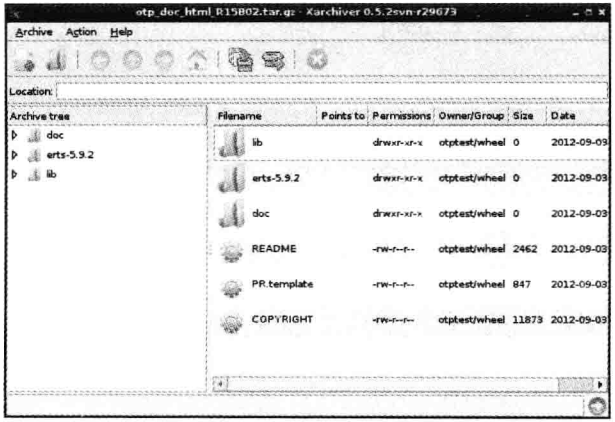


图 2-16 XArchiver 的运行截图



图 2-17 Education 菜单

Scratch

Scratch 由麻省理工学院媒体实验室（MIT media labs）Mitchel Resnik 领导的 Lifelong Kindergarten 小组开发。其目标是让人们更轻松地学习编程入门知识，这让我们介绍 Scratch 也很轻松。

我们并非 Scratch 方面的专家（事实上我们也是拿到树莓派后才开始接触它的），但我们相信像 Scratch 这样的优秀应用会通过树莓派的普及而大放异彩。不需要全面地掌握各种细节知识，人们就可以通过 Scratch 提供的系统实践编程。它容易上手也很强大，不相信的话，看看图 2-18 吧，很有趣不是吗？

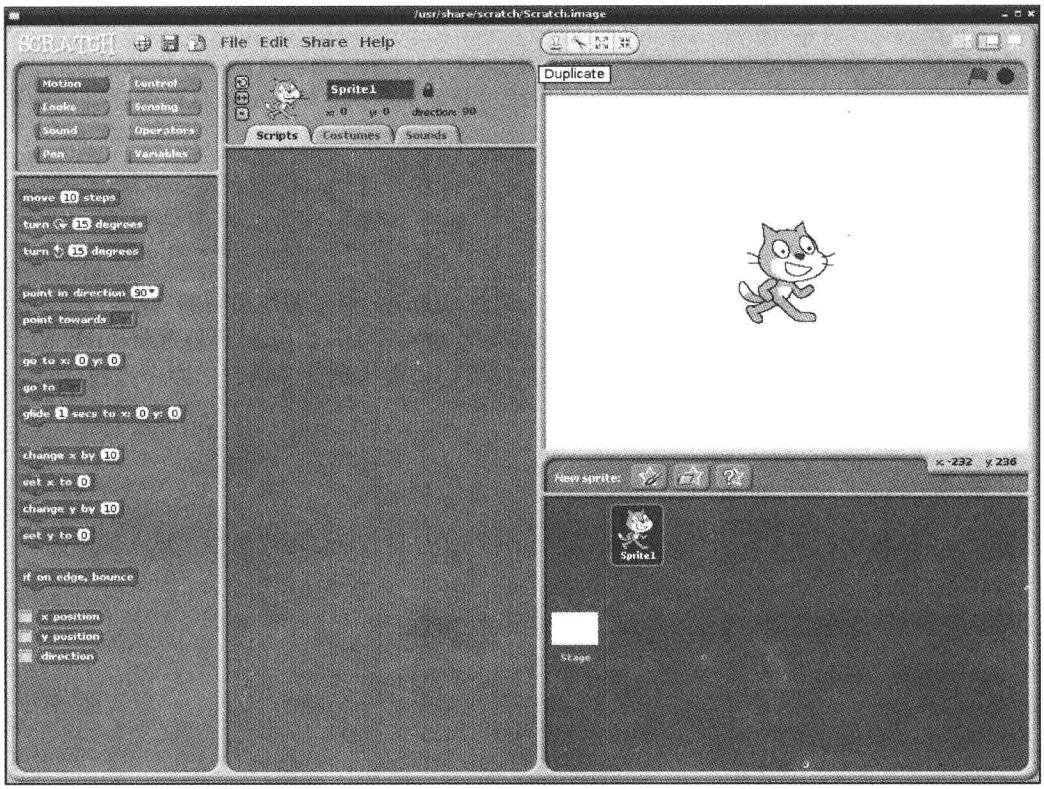


图 2-18　Scratch 截图

Squeak

Squeak 是一种预装在树莓派上的编程语言。由于许可证的原因，其安装并不完整，无法直接使用。这或许是个暂时性的问题，可能会在未来的 Raspbian 中得到修复。目前，如果打开 Squeak，你会看到如图 2-19 所示的对话框。

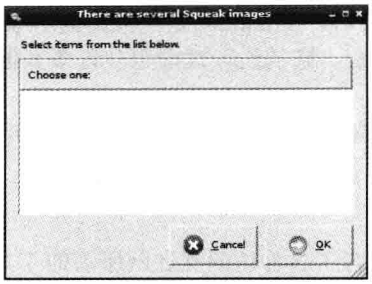


图 2-19 初次运行 Squeak

在树莓派官方论坛上可以找到解决方案，通过该方案我们可以完成 Squeak 的完整安装。打开 LXTerminal，然后依次输入以下命令（不妨在阅读完第 3 章对命令行工具更加熟悉后，再回过头来看看这些命令）：

```
$ wget -r -O SqueakV41.sources.gz http://ftp.squeak.org/4.1/SqueakV41.sources.gz
$ gunzip -f SqueakV41.sources.gz
$ wget -r -O Squeak-4.3-All-in-One.zip http://ftp.squeak.org/4.3/Squeak-4.3-All-in-One.zip
$ unzip -j -o Squeak-4.3-All-in-One.zip "Squeak-4.3-All-in-One.app/Contents/Resources/Squeak4.3.*"
$ sudo mv  SqueakV41.sources /usr/share/squeak
$ sudo mv  Squeak4.3.image /usr/share/squeak
$ sudo mv  Squeak4.3.changes /usr/share/squeak
```

你现在或许还不明白每行命令的具体含义，简而言之，它们都是用于下载并安装 Squeak 缺失的模块。执行过这些命令后你就能正常地使用 Squeak 了，如图 2-20 所示。

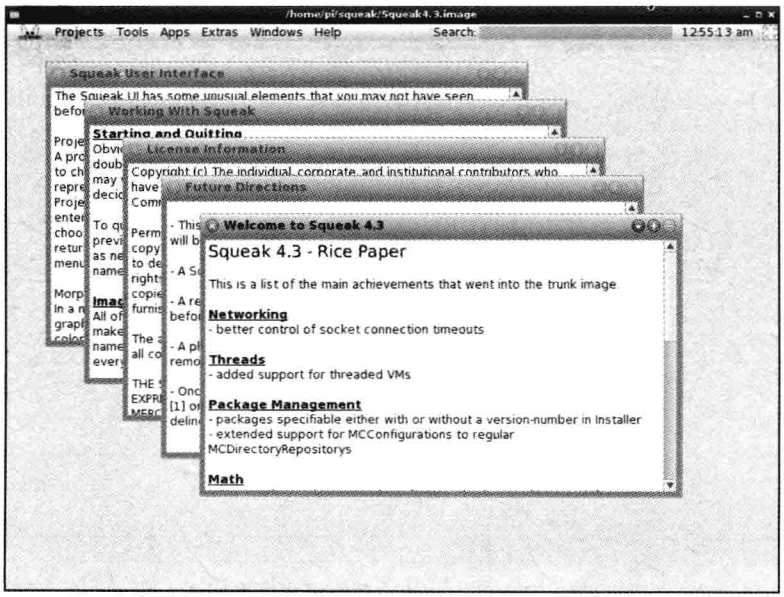


图 2-20 Squeak 正常运行截图

Squeak 是一个 Smalltalk 的变种，Smalltalk 是一种面向对象、支持反射的编程语言，它有着简捷高效的语法，多年来经久不衰。它的设计理念也被许多其他编程语言借鉴，包括 Ruby 和我们刚刚提过的 Scratch。

在我们看来，Squeak 是一种有趣有用的编程语言，通过它你可以开始在编程世界的有趣旅行。

互联网

Internet（互联网）菜单中包含了各种访问互联网所需的工具，如图 2-21 所示。



图 2-21　Internet（互联网）菜单

Raspbian 预装了 3 种不同的浏览器。这几个浏览器都崇尚极简主义的设计，不妨在试用后选择一个喜欢的。Midori 是我们的最爱，它非常容易上手。Midori 贴心地提供了标准模式和隐私模式供用户选择。需要注意的是，与不少浏览器模式切换时改变主题或图标的方法不同，Midori 将 Private Browsing（隐私模式）的提示直接写在了标题栏上，如图 2-22 所示。在 Private Browsing 模式下，无法使用添加书签以及重新打开已经关闭的标签页这两项功能。经常使用这一模式的用户记得三思而后行（因为浏览器不会保留任何历史记录信息）。

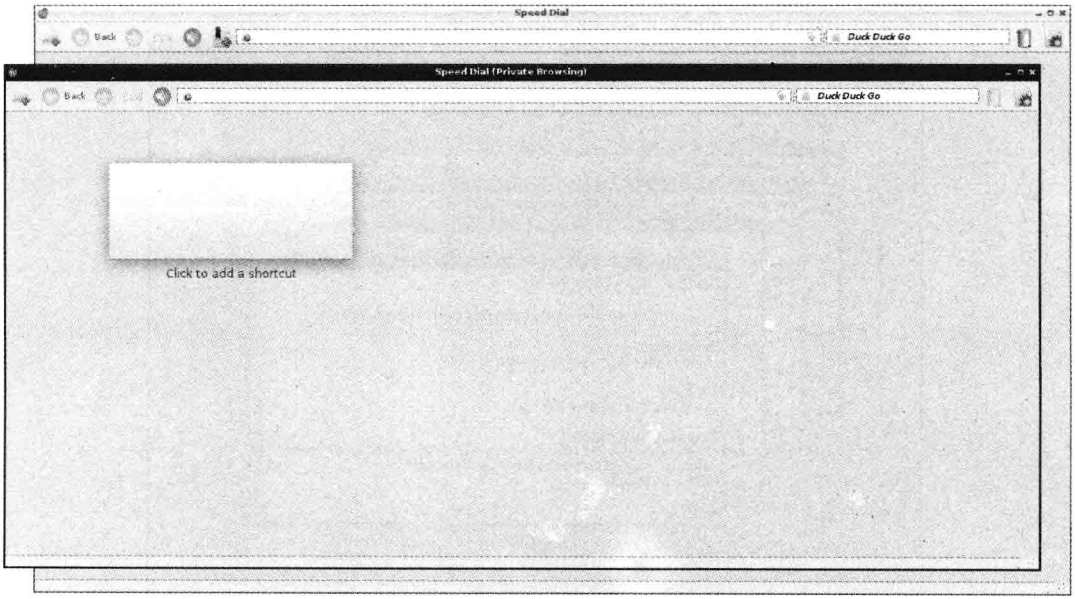


图 2-22　Midori 的隐私模式和标准模式

编程



图 2-23 Programming（编程）菜单

Programming（编程）菜单中共有 4 个应用，其中两项在 Education 菜单中我们已经介绍过了，如图 2-23 所示。余下的两项是 IDLE 和 IDLE3，它们都是用于编写 Python 的集成开发环境（IDE）。集成开发环境主要是用来提供一站式的编程环境，提供了运行、测试、高亮显示代码等功能。之所以这里提供了 IDLE 和 IDLE3 两个版本，是因为 Python 的两个主要版本互不兼容。虽然很多情况下在 Python2.x 下写的脚本在 Python3.x 下也可以运行，但这两个 Python 版本间确实存在无法弥合的兼容性问题。这也是人们需要同时安装两个不同版本 Python 的原因。

注意 在本书写作时，Python2.x 仍然是大多数开发者的选择，应用广泛。这是由于 Python2.x 拥有大量的库可以使用，相关的文档、博客及网络社区资源也非常丰富。另一方面，Python3.x 相比上一个版本进行了大量的改进，并仍在不断进步中。总体来说，Python3.x 是未来的方向，如果你刚刚开始学习编程，选择它总不会错。当你真正不得不用到 Python2.x 时，会发现并不需要太多的学习成本。

两个版本的 IDLE 在使用方法上十分相似，图 2-24 是 IDLE 的截图。

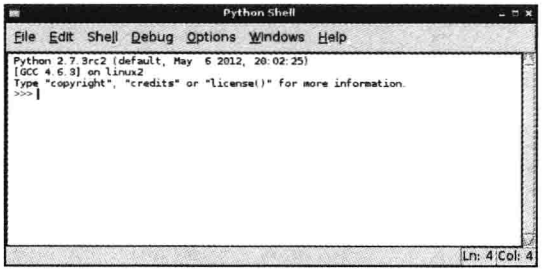


图 2-24 IDLE 启动界面

首次启动 IDLE 时，会打开 Python Shell，这和在终端中输入 python 命令看到的没有什么不同。IDLE 的真正价值在于提供了可以高亮显示 Python 代码的编辑器。代码高亮这一概念对编程新手或许会有些陌生，这是指通过不同的颜色显示代码的不同元素来提高代码的可读性。比起白底黑字的显示，高亮代码中的关键字和结构后，检查代码中存在的错误更加容易。还有点迷糊？打开 IDLE 试一试吧，你会喜欢它的。图 2-25 中的程序会在屏幕上打印“Hello world!”20 次，然后退出。

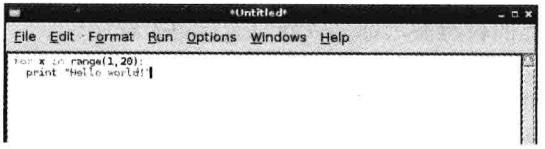


图 2-25 使用 IDLE 的 IDE 组件

可以看到，IDLE 还支持代码高亮，代码高亮可以更方便地理解代码的结构层次。Python 是一种强大的编程语言，它的应用极其广泛。如果想更深入地学习 Python，不妨登录 www.python.org 获得更多信息。

系统工具

System Tools（系统工具）菜单中只有一个应用，如图 2-26 所示。



图 2-26　系统工具菜单

Task Manager（任务管理器）可以显示出机器上正在运行的所有进程以及系统资源占用情况。图 2-27 是在树莓派上运行 Task Manager 的截图。

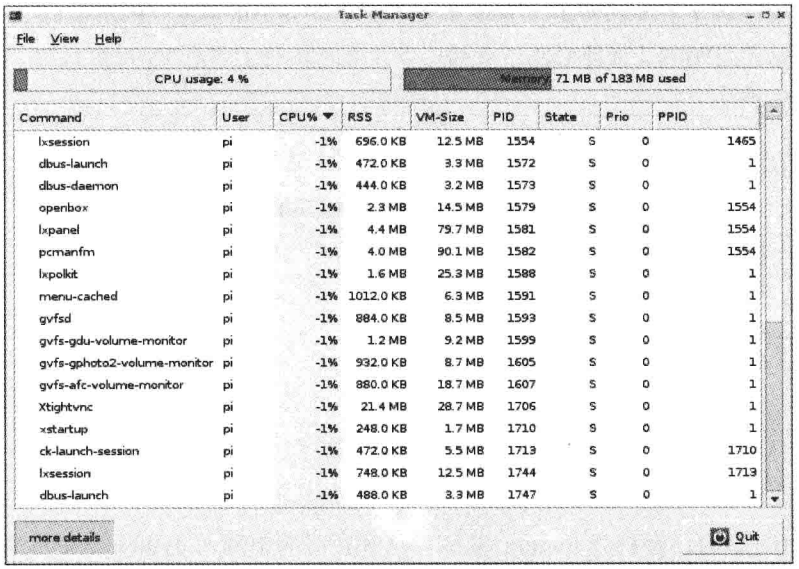


图 2-27　任务管理器

可以看到，图 2-27 中 CPU 正处于空闲状态。考虑到 CPU 运行着一个完整的操作系统，CPU 如此空闲是绝不常见的。而树莓派使用的 ARM 处理器与 Intel 的高端处理器相比差距不小。这正说明通过精心的设计，我们可以通过极少的资源实现足够好的桌面环境。

内存的情况也是如此。我们打开了一个 VNC 远程桌面，可以看到内存的占用维持在 50%以下。如果使用 512MB 内存的树莓派，你可以同时运行更多程序。

通过进程列表你可以清晰地看到有哪些程序正在运行以及它们占用了多少资源。当你的树莓派运行得不太流畅时，你可以查看是否有哪个进程占用了过多的系统资源。如果真地有这样贪心的进程，你可以右键单击它，选择 Kill（结束进程），帮助树莓派恢复正常。

设置

好了，接下来让我们来看看如何让你的桌面显得更个性。Raspbian 提供的 Preferences（设置）工具十分简单易用，这里我们主要介绍其中的两个部分，余下的供读者自己尝试。改变这些设置不会破坏系统，所以，请随意折腾吧。

桌面设置

右键单击桌面空白处，选择 Desktop Preference（桌面设置），界面如图 2-28 所示。

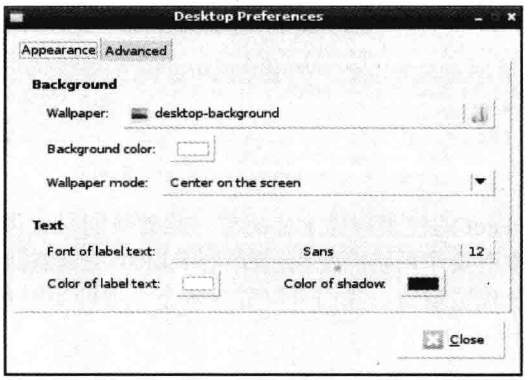


图 2-28　桌面设置

可以看到，在 Appearance（显示）标签页中所有选项都简单明了，读者可以根据喜好随意设置。由于 Advanced（高级）标签页并不常用，本书不做介绍。

自定义界面外观

你如果想自定义整个桌面，设置主题、图标以及字体等界面外观元素，你可以通过 Preferences（设置）菜单访问这个工具，如图 2-29 所示。

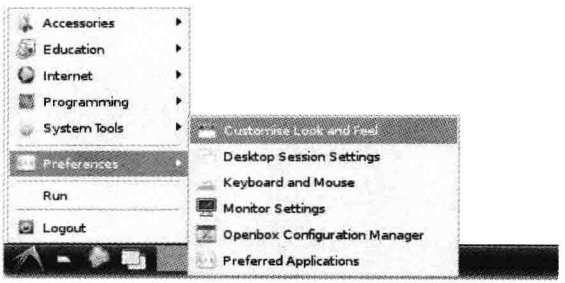


图 2-29　Preferences 菜单

Customize Look and Feel（自定义界面外观）打开后如图 2-30 所示。

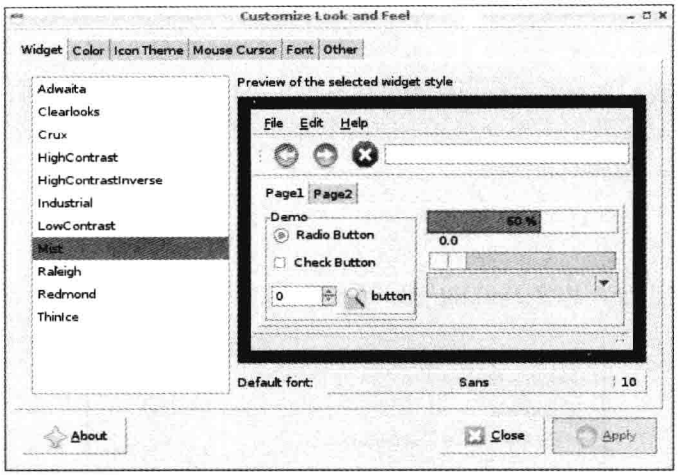


图 2-30　Customize Look and Feel

Customize Look and Feel 的使用同样十分简单。此处我们只看第一个标签页，在这里你可以通过一次鼠标单击改变整个操作系统的配色。默认的 Mist 主题看起来就不错，但系统也为你提供了不少其他选择。通过预览窗口，我们可以看到各个主题的实时预览效果。

本章小结

好了，我们已经走马观花地了解了整个桌面环境。或许有些简单，但我们已经为进一步的探索打下了基础。请记住树莓派存在的意义就在于帮助人们探索各种和计算机有关的有趣课题，别害怕失败，开始折腾吧！

本章中我们浏览了开始菜单中预装的各种应用，接触了一些 Linux 常见的交互界面，还介绍了两项最常用的系统个性化设置。

在下一章中我们将开始介绍更为重要的内容，是时候接触一下命令行工具了。熟悉掌握了命令行工具后，总有一天你会想知道没有命令行的那些年你是怎么活下来的。

第 3 章

驾驭命令行

本书一般不会从一张图片开始新的章节，本章是唯一的例外。这里我们向逝去的经典设计致敬，看看图 3-1 所示的这张照片吧（来自维基百科）。

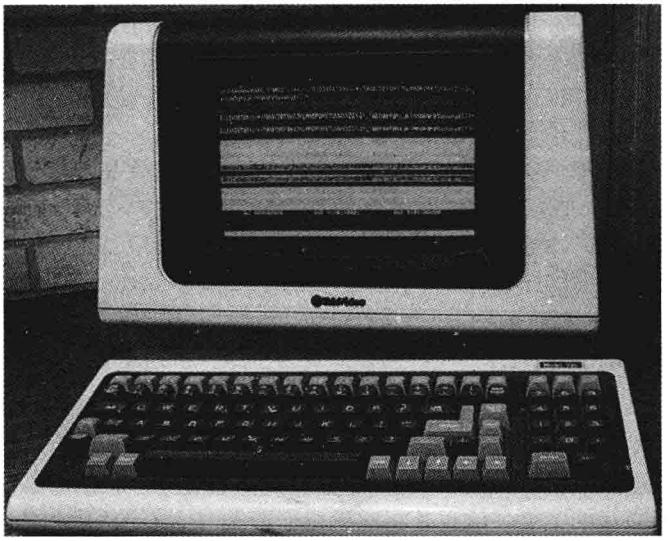


图 3-1　TeleVideo 925 终端机

你或许会好奇为什么我们会向你展示这个诞生于 20 世纪 80 年代（准确地说是 1982 年年末）的技术。那个时代的计算机远不如现在的光鲜亮丽，它们甚至还没有配备彩色显示器和鼠标，屏幕上所能看到的只是黑底绿字的命令行界面。那时距离鼠标的普及尚需时日，人们只能通过文本菜单和命令行工具与计算机进行交互。在 30 年前的当时，命令行工具以其远优于文本菜单的灵活性，成为了最常用的交互方式。那时的计算机用户在图中这种被称为“终端机”的机器前废寝忘食地敲打着各种命令。

本书的第二部分将着重讨论如何驾驭命令行工具。虽然在过去的几十年间，计算机似乎发生了翻天覆地的变化，但是万变不离其宗，最后人们发现命令行依旧是十分强大的工具。

本章中，我们还会简要介绍一下计算机交互方式的发展历史，以及为什么命令行工具至今仍

有强大的生命力。最后我们会介绍几种可供选择的终端以及它们的安装、设置。我们也会讲解命令提示符的作用以及有关 Shell 的一些小知识。

3.1 曾经的计算机

今天我们只需要花 25 美元就可以买到计算机，而在 20 世纪 70 年代，任何具有正常计算能力的计算机都需要上万倍的花费。那时的计算机体积巨大，需要占据整个房间乃至整个楼层的空间。同时，其能耗也十分惊人，作为那个时代的亲历者我们至今记忆犹新。简而言之，那个时代计算机的关键问题在于其过于巨大的体量。

当一台计算机的价格比整座机房大楼的价格更高时，计算机的所有者显然希望能榨干它的每一滴计算能力。人们尝试通过时分复用的方法来提高计算机的使用效率，这一方法的出发点在于在人机交互过程中，计算机的速度比使用者快得多。虽然当年计算机的计算能力在今天已经不值一提，不过在当时，人们已经十分惊叹于它们的计算能力。这就引出了这样一个问题，既然大多数时间计算机只是在等待使用者输入指令和数据，为什么不让它在等待的时候做些别的事呢？

不久人们就发现当计算机在等待输入时，多人同时录入数据更有效率。逐渐的，录入速度就不再是计算机运行效率的瓶颈了。

这是个好主意，但该如何实现呢？ 我们至少需要让每个需要和计算机交互的人面前都有台输入设备，而在那时在每人面前都放一台基本的计算机也是相当不现实的。人们注意到，其实不需要那么多台计算机，只要让每个人都有和同一台大型机交互的方法就可以了。实际上，只需要有键盘、屏幕以及足够长的线路，整套系统就可以工作了。

和简易终端打个招呼吧

就像上面说的，键盘、屏幕以及网线就构成一台简易终端，如此简陋的配置也是它被称为简易终端的原因。通过从大型机上传来的指令，简易终端将文本绘制到屏幕上。当按下键盘按键时，简易终端会将信息发送至大型机，并根据返回信息更新屏幕显示。简易终端不具有任何计算能力，它只是一种与计算机交互的工具。这也使得它价格低廉并易于安装和维护。

随着经年的技术发展，简易终端不再那么“简易”了。人们为它添加了彩色显示和亮度控制功能，甚至还加入了屏幕拉伸及屏幕管理等功能。但简易终端这种与计算机的交互概念并没有太多的改变。事实上，虽然简易终端在现代网络发展及计算成本降低的浪潮中逐渐被淘汰，但新的系统仍沿用了其设计理念。这不只是出于兼容性的考虑，更是因为简易终端具有许多至今仍十分实用的特性，这也是为什么我们能见到的所有终端都有类似的界面。

GUI 和 TUI

你可能常听人提起 GUI 和 TUI 这两个概念，本书也会多次提及。幸运的是，这两个概念十分简单且易于理解。

GUI（Graphical User Interface，图形用户界面）被所有现代计算机系统所采用。起

先我们可以通过是否可以用鼠标操作来鉴别 GUI。但近年来这条鉴别的界限有些模糊了，随着 iPad 和 Android 手机之类的普及，你也可以直接通过手指来操控 GUI 了。TUI（Text-based User Interface，文本用户界面）通常用于与命令行工具交互。TUI 可提供基础的菜单系统，通过键盘而非鼠标控制。

简而言之，没有鼠标且只能用键盘控制的是 TUI。如果可以用鼠标或手指单击并可以浏览多媒体文件的就是 GUI。

现代终端

简易终端经过了不断的进化，在我们今天使用的 Linux 系统中重生了。比如，我们常常通过网络远程登录你的计算机（或者树莓派）。当远程登录树莓派时，你需要在一台计算机上运行一个客户端，并且在树莓派上运行一个服务器，来接受连接。一但连接成功，你就可以“远程控制”你的树莓派，这样的连接被称为“远程终端”。这里需要注意的是，当处于远程连接状态时，你输入的所有指令都是直接在树莓派上执行的。这就好比通过电视机遥控器和电视机上的按钮换台，结果是完全相同的。

终端的故事到此告一段落了，让我们回到现代来看看连接到树莓派上的远程终端长什么样子吧，如图 3-2 所示。

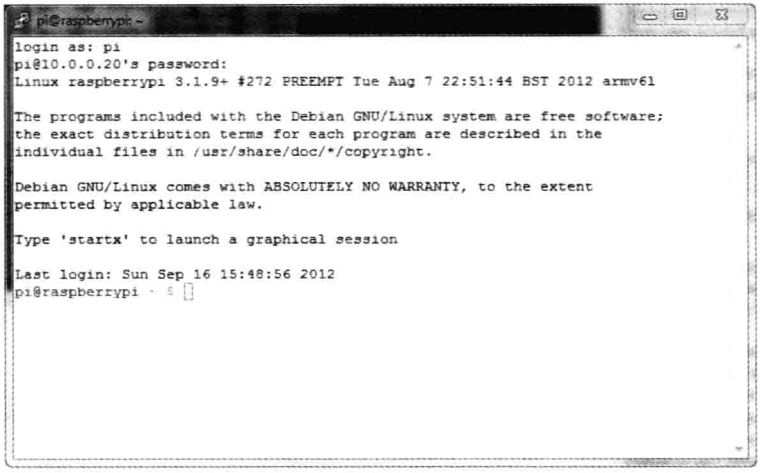


图 3-2 连接到树莓派上的 SSH 连接

可以看到这是一个典型的文本用户界面，程序不是运行在本机上，而是网线另一端的树莓派上。远程终端具有文字高亮显示等诸多贴心的特性。这里我们使用 SSH（Secure Shell，后文会有详细讲解）连接到树莓派上。

为什么我们还要学习命令行工具呢？

这是个好问题，在这个 iPhone、iPad 以及 Android 等大量触控设备大行其道的时代，为什么

还要学习命令行呢？对大多数人来说，命令行工具听起来有些过时，让人提不起兴趣。然而，作为命令行工具的忠实拥簇，我们认为花费时间学习命令行是绝对值得的，理由如下：

简洁

简洁的设计是我们学习命令行工具的一个重要原因。没有哪种设计能比只进行文本收发的文字界面更加简洁。或许你不能用它查看假期里拍的照片，但却可以通过它清爽的界面控制你的树莓派。

快捷

这听上去或许有点难以置信，但命令行工具确实能帮助你高效快捷地完成各种任务。这是因为图形界面下需要许多次点击、跋山涉水才能完成的操作，往往一行 Shell 命令就能精确高效地完成。

轻量

如果你有过通过很慢的网络访问远程桌面的惨痛经历，我们下面的观点很可能会引起你的共鸣。通过网络传输屏幕图像（特别是高分辨率的彩色图像）无法避免地要占用大量带宽。数据传输的延时也无法避免，这就导致访问远程桌面时的卡顿感。

只传输文本信息的远程终端不存在这一问题，它最初就是为带宽极小的系统设计的。所以即便是缓慢的拨号连接，远程终端也可以很好地工作。

相比于远程桌面，远程终端对服务器端的压力也小得多。图形界面的使用需要占用大量系统资源，例如，系统需要跟踪鼠标指针以确保用户单击窗口或按钮时能够正确反应。维护图形界面的成本远比想象得要高。鉴于此，Linux 服务器上往往不安装图形界面，即便安装了也很少有人去使用。

强大

命令行工具让你具有超级的能力。它让你能够以图形界面下无法做到的方式实现复杂的想法。比如，你可以将许多命令串在一行中使用，前一个命令的输出会作为后一个命令的输入。这种方法称为“管道”（pipes，第 5 章会详细讲解这一概念）。

随时可用

虽然近年来有所改善，但图形界面仍是 Linux 系统中稳定性最差的部分。你显然不能指望当服务器宕机后需要恢复时通过图形界面操作。

相比之下，终端才是和 Linux（以及其他类 UNIX 操作系统）交互最为自然的方式，它总在那里，不离不弃。学会驾驭命令行工具后，你能够轻松摆脱以前种种令人抓耳挠腮的困境。

还在犹豫吗？

我们希望上面这些讲解能让你多少对命令行工具产生些兴趣。如果你还在犹豫，这也很正常，命令行工具的学习相对来说要困难一些，但是一旦经过了这个阶段，你就会感受到它的强大能力。后面的章节，我们会开始尝试实际使用命令行工具，了解其具体使用细节。通过一些基础命令的学习，你将踏上自己的折腾之旅。

终端和虚拟终端

大多数情况下终端和虚拟终端这两个概念可以混用。实际使用中，人们通常是通过Linux或SSH使用虚拟终端。咬文嚼字地来讲，终端指的是某种物理设备，例如连接在计算机上的键盘和显示器，用户通过终端设备直接在命令行中操作计算机。而虚拟终端指的是运行在一个操作系统中的虚拟设备，例如在Linux中，每个虚拟终端都对应一个虚拟设备。

虽然这里介绍这两个名词的不同之处，但实际上我们已经很难接触到真正的终端了，我们在日常使用中，都主要使用虚拟终端。

3.2 3种终端

打开命令行有3种主要的方法。第一种是坐在计算机前，直接使用控制台；第二种是使用图形化界面中的虚拟终端；最后一种是使用网络连接SSH终端。

控制台

单纯从文字上看，控制台（console）和终端（terminal）是一个意思，不过我们这里说的控制台指的是用户坐在计算机前，直接使用连接的键盘和显示器登录系统时，所使用的简单的文字界面。

这种方法一般仅限于系统管理员（也就是root用户），平时服务器都锁在专门的机房中，以保证安全，所以只有系统管理员能够直接接触到这些计算机。

现在主流的发行版几乎都不再支持这种使用方式了，远程网络连接的管理方式更为普遍（甚至有的时候，管理员都不能直接接触到计算机了），所以控制台的使用也越来越少。

不过，我们还是能在树莓派上使用一下控制台，启动树莓派后，按住Ctrl和Alt键，并按下F1到F6之间的任一个键，就能够切换到控制台了。F7键对应的则是默认启动的图形化界面，所以按住Ctrl和Alt键，并按下F7键，即可回到图形化界面。

当你无法通过网络远程登录树莓派或者图形化界面出现了问题时，你可能就需要使用控制台来修复问题了。不过，一般情况下我们可能很少会使用控制台，而且我们还有其他的替代方式（下一节我们将会介绍），所以除非图形化界面不可用了，或者你的确更喜欢控制台界面，我们一般不会使用它。

我们下面还将介绍另外两种命令行使用方法，这两种方法更加适于日常使用，而且也更加容易启动。

在图形化界面中虚拟终端

Raspbian 系统中使用 LXDE 图形化桌面系统，这款轻量级图形化桌面的设计宗旨就是简洁易用。

其自带的终端名为 LXTerminal，并且是系统默认安装的，其界面如图 3-3 所示。

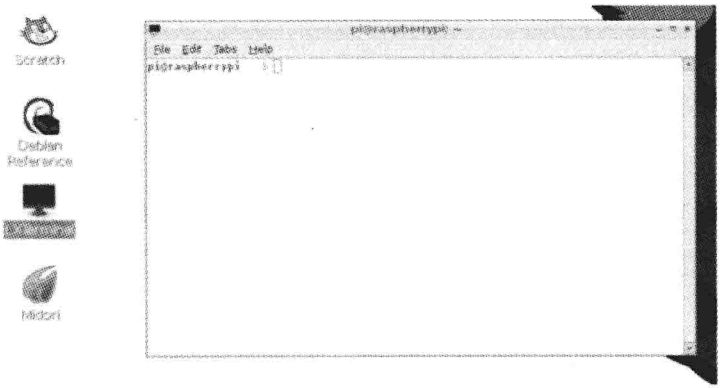


图 3-3 启动 LXTerminal

我们双击图标启动 LXTerminal，你会发现这个终端真的非常简单。如果你将树莓派当作普通的计算机来使用，这个控制台能够满足你的需求。不过如果你想远程登录并使用树莓派的话，你还需要阅读下一节。

使用 SSH 远程连接

使用 SSH 通过网络远程连接树莓派是一种很安全的做法（参考“SSH 为什么这么特别”），SSH 可以提供所有虚拟终端的功能，同时你还可以远程登录，而不需要坐在计算机前。比如，你可以在公司的计算机上，安全地远程登录家中的树莓派。很多人将树莓派作为服务器主机来使用，所以我们觉得 SSH 连接是最好的使用方法。

SSH 的连接需要两个程序一起完成，分别是 SSH 服务器和 SSH 客户端，不过这两个都是自由软件，而且设置非常容易。

SSH 为什么这么特别

SSH 是事实上的 UNIX 系统远程连接标准，几乎所有的联网设备都支持该协议，包括打印机和高级的交换机。SSH 之所以流行，是因为该协议对客户端和服务器之间的所有通信都进行了加密，如果有人试图监听你的连接，将会看到无法处理的数据，SSH 还支持对连接的服务器进行验证。

在SSH出现前，人们普遍使用的是telnet，telnet是一个非常简单的协议，而且不支持加密，所以如果有人监听你的通信，就能很容易得到明文传输密码等重要信息。telnet也无法验证服务器是否是你的计算器还是别人伪造的。

你仍然可以在Linux上安装并使用telnet，但是我们并不建议你这么做，如果可以，还是尽量使用SSH进行远程连接。

在树莓派上设置SSH服务器

在树莓派上不用设置即可使用SSH，除非你安装时Raspbian选择了禁用SSH。Raspbian中的SSH服务器随机器一起启动，所以无需任何设置就能接受SSH连接。

不过连接前，我们还需要使用一下图形化界面中的虚拟终端，查看树莓派当前的IP地址。树莓派支持DHCP（Dynamic Host Configuration Protocol，动态主机设置协议），所以当你将树莓派接入常见的网络中，它将会自动配置网络地址，无需你手工设置。

在虚拟终端中输入sudo ifconfig（interface configuration的缩写）命令可以查看IP地址，我们将会看到如图3-4所示的内容。

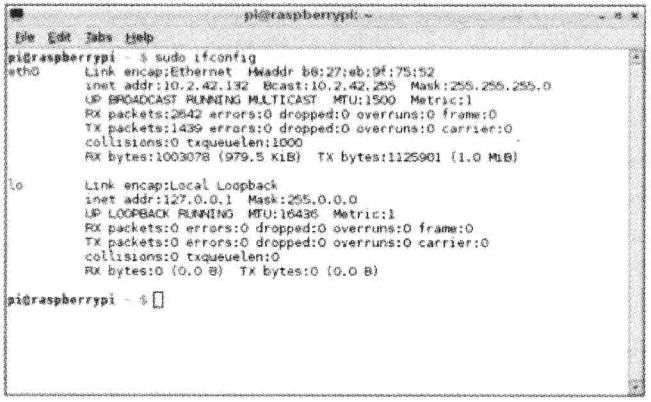


图3-4 在树莓派中执行ifconfig

我们看到了两个网络接口。其中一个是本地回环设备（缩写为lo），这是一个虚拟网络接口，用于本地应用程序之间的通信。由于不是真正的硬件，所以该设备速度快，效率高，而且始终可用，在所有计算机上的配置都一样。在Linux系统中，许多系统服务（例如打印服务和图形化界面）都使用该设备实现客户端/服务器架构。

了解了本地回环lo设备后，我们接下来介绍树莓派上真正的物理网卡设备eth0（Ethernet device number 0的缩写）。本书不会详细介绍配置的每一个细节，我们只需要从ifconfig中得到IP地址即可。从图3-4中，我们看到如下的内容：

```
inet addr: 10.2.42.132 Bcast: 10.2.42.255 Mask: 255.255.255.0
```

其中显示了我们的树莓派IP地址为10.2.42.132，有个这个地址，我们就能连接树莓派了。

选择一个 SSH 客户端

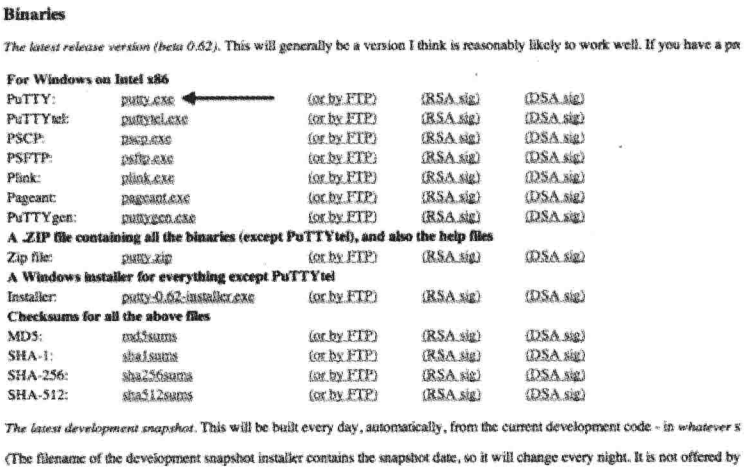
如果你准备在 Windows 或者 Mac 计算机上使用 SSH 连接，本节的内容正是你所需的。当然如果你使用其他操作系统也没关系，一定有适合的 SSH 客户端程序。使用各种 SSH 客户端连接树莓派的主要步骤都差不多，所以应该不会影响你阅读本节的内容。

PuTTY（Windows 系统）

和其他 SSH 客户端相比，PuTTY 可以算是标准的 SSH 客户端，它不仅有完整的功能，而且还是自由软件。你可以从以下地址下载最新版的 PuTTY：

```
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.
```

输入整个地址可能比较麻烦，你可以在 Google 中搜索“Download PuTTY”，并在结果中访问该页面（见图 3-5），单击第一个链接，下载 putty.exe。



Binaries

The latest release version (beta 0.62). This will generally be a version I think is reasonably likely to work well. If you have a pr

For Windows on Intel x86				
PuTTY:	putty.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYtel:	puttytel.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSCP:	pscp.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSFTP:	psftp.exe	(or by FTP)	(RSA sig)	(DSA sig)
Plink:	plink.exe	(or by FTP)	(RSA sig)	(DSA sig)
Pageant:	pageant.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYgen:	puttygen.exe	(or by FTP)	(RSA sig)	(DSA sig)
A .ZIP file containing all the binaries (except PuTTYtel), and also the help files				
Zip file:	putty.zip	(or by FTP)	(RSA sig)	(DSA sig)
A Windows installer for everything except PuTTYtel				
Installer:	putty-0.62-installer.exe	(or by FTP)	(RSA sig)	(DSA sig)
Checksums for all the above files				
MD5:	md5sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-1:	sha1sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-256:	sha256sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-512:	sha512sums	(or by FTP)	(RSA sig)	(DSA sig)

The latest development snapshot. This will be built every day, automatically, from the current development code - in *whatever s*

(The filename of the development snapshot installer contains the snapshot date, so it will change every night. It is not offered by

图 3-5 PuTTY 下载页面

PuTTY 不需要安装，也无需管理员权限，直接运行下载的 putty.exe 即可，你还可以把它拷贝到 U 盘中随身携带。当然，你也可以下载 PuTTY 安装包（Installer），安装后在桌面和开始菜单中都会有 PuTTY 的快捷方式。

双击图标运行 PuTTY，你有可能会看到如图 3-6 所示的对话框。

Windows 系统警告你该程序没有经过数字签名，你也许之前也见过这个对话框，这个警告有助于提高计算机的安全性。不过 PuTTY 的官方版本的确没有数字签名，所以你可以忽略这个警告，单击 Run（运行）。如果你经常使用 PuTTY，可以考虑取消选中“打开这种类型的文件前始终询问”（Always ask before opening this file），否则每次运行前，Windows 都会询问你。

单击 Run（运行），你应该会看到如图 3-7 所示的连接对话框。

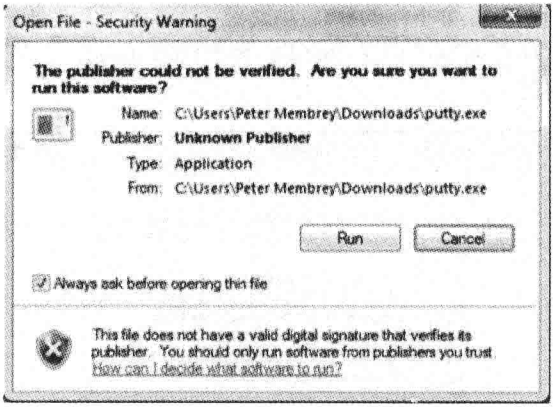


图 3-6 Windows 安全警告

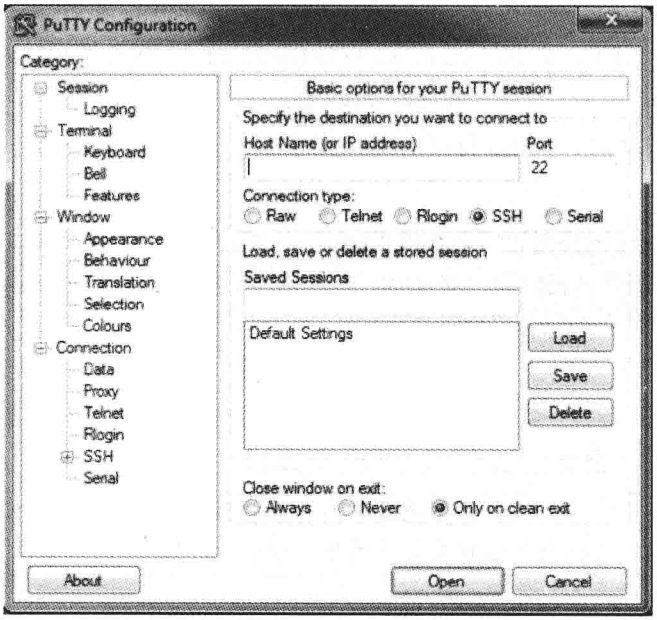


图 3-7 启动 Putty

你只需要在 Host Name（or IP address）（主机名或 IP 地址）处输入树莓派的 IP 地址，其他设置保持默认值即可，单击 Open（打开）。由于是第一次连接，所以 PuTTY 会弹出如图 3-8 所示的警告对话框。

这是一个 SSH 安全警告，不过不用紧张，出现这个警告是由于 PuTTY 之前从来没有连接过你的树莓派，所以 PuTTY 向你确认是否信任树莓派提供的 SSH 密钥，单击 Yes 即可。SSH 密钥用于确认 IP 对应的服务器没有发生变化，PuTTY 记住密钥后，再次使用该 IP 连接对应的服务器时，它会验证 SSH 密钥是否相同。如果服务器的密钥发生了变化，PuTTY 会警告你。正常情况

下应当不会再次出现这个警告，除非你的操作导致密钥发生变化，例如重装了系统或者新买了一台服务器。如果你没有做什么调整，但是 PuTTY 警告你密钥发生了变化，这时你需要注意一下服务器是否出现了什么问题。

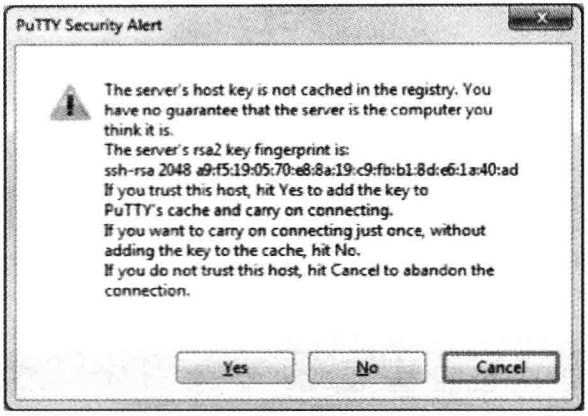


图 3-8 Putty 警告对话框

PuTTY 连接到树莓派后，就会显示出很熟悉的黑底白字命令行界面，并显示：

```
login as:
```

我们不准备为 SSH 连接设置不用的用户账户，所以输入 pi 并输入密码，按回车键。我们这里提醒一下，pi 用户的默认密码是 raspberry，如果你之前没有修改密码的话，使用这个默认密码即可。完成后你应当会看到以下命令提示符：

```
pi@raspberrypi ~ $
```

到这里已经完成了连接，如果你不使用 Mac 计算机（或者不打算使用 Mac 计算机连接树莓派），那你可以跳过下一节。

注意 当使用远程连接时，很容易会搞混不同的远程计算机，所以如果你做一些操作时，比如重启树莓派，请确认在正确的远程连接中执行了命令。

SSH（Mac 系统）

如果你使用的是 Mac 计算机，那就不需要使用 PuTTY 了。Mac 计算机的操作系统符合 UNIX 标准，所以自带了 SSH 客户端，你只需要打开终端（Terminal）并输入 ssh 命令。

在应用程序（Applications）目录的实用工具（Utilities）中，打开终端（Terminal），或者你也可以在 Spotlight 中输入 Terminal，进行搜索。

Mac 系统的终端和你在树莓派上见到的差不多，我们说过 Mac 计算机的操作系统也是符合 UNIX 标准的，所以很多 Linux 命令可以直接在 Mac 的终端中使用。

打开终端，连接好树莓派，输入 ssh 命令，开始连接树莓派，整个过程如图 3-9 所示。

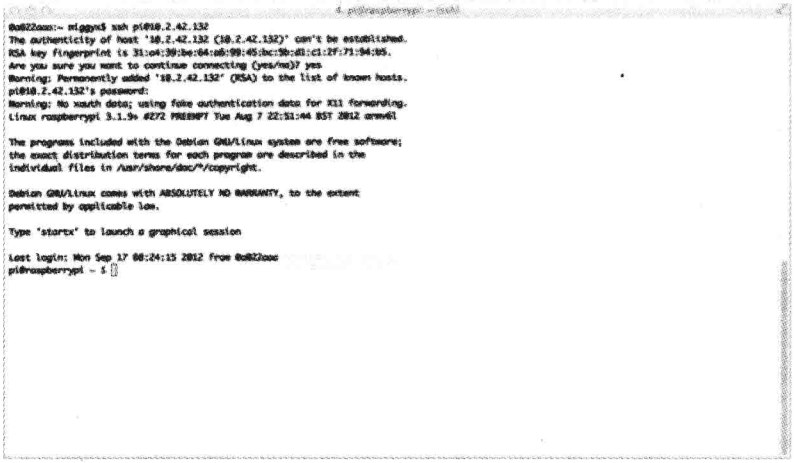


图 3-9　在 Mac 上连接树莓派

下面我们来详细介绍整个连接过程。由于没有图形界面，你需要在命令中输入相关的参数，完整的命令如下所示。

```
$ ssh pi@10.2.42.132
```

我们在 ssh 命令参数中提供了树莓派的用户名和 IP 地址，整个命令可以理解为“使用用户名 pi 连接位于 10.2.32.132 的树莓派”，其中的参数看起来有点像 E-mail 地址，所以很好理解。你的树莓派应该会使用一个不同的 IP 地址，那么替换这个命令中的 IP 地址即可。

按回车键后，SSH 会提示你连接到了一台新计算机。

```
The authenticity of host '10.2.42.132 (10.2.42.132)' can't be established.
RSA key fingerprint is 31:a4:39:be:64:a6:99:45:bc:5b:d1:c1:2f:71:94:b5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.2.42.132' (RSA) to the list of known hosts.
pi@10.2.42.132's password
```

这个 SSH 警告表明第一次连接到你的树莓派，并显示它提供的 SSH 密钥，这里你可以忽略警告，输入 yes 即可。这个警告的具体原因我们在 PuTTY（Windows 系统）一节已经做了介绍，你可以阅读该节进一步了解。之后 SSH 提示你输入密码（pi 用户的默认密码是 raspberry），输入完成后，你应该会看到以下命令提示符。

```
pi@raspberrypi ~ $
```

至此，我们已经能够通过 SSH 连接到树莓派。

3.3　欢迎来到命令行的世界

你可以通过多种方式连接到树莓派，不过最终你都会得到类似下面的命令提示符。

```
pi@raspberrypi ~ $
```

命令提示符提供了很多的信息，阅读到这里，你可能也已经理解了一些，不过我们将在这节详细说明。

命令提示符的第一部分（本例中的pi）是用户名，在这个目前只有一个用户的树莓派上，用户名似乎是固定的，但当你需要管理多台计算机，使用多个不同的账户时，用户名可以让你不会混乱。命令提示符的第二部分（本例中@符号之后的raspberrypi）是当前连接计算机的主机名，结合第一部分的用户名，你可以完全了解当前的连接状态。

注意 当你操作多个终端时，很容易会出现错误，有时你可能在错误的计算机上执行了多条命令后，才发现自己搞错了。虽然多数时候都不会造成大问题，不过你也有可能一不小心重启了一台重要的服务器，而不是你的树莓派。所以在终端连接中操作一定要小心。

命令提示符的下一个部分（本例中的~，波浪号）告诉你当前的工作目录，波浪号表示当前用户的主目录（home directory）。波浪号可以当作是一个捷径，因为你会经常进入主目录（比如树莓派的/home/pi），例如：

```
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $ cd ~
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $
```

本例中，我们使用pwd命令（print working directory的缩写）显示当前工作目录的完整路径。可以看到该命令打印出/home/pi，也就是pi用户的主目录。我们通过cd命令（change directory的缩写）进入~目录之后，再使用pwd命令，我们发现还是在相同的目录。我们会在下一章详细介绍这些命令，所以我们先接着向下介绍命令提示符。

因此~可以读作主目录，或者在本例中，详细点说是pi用户的主目录。所以可以不用再将它读做波浪号了。

不过有的时候，你也需要将其读作波浪号。例如向其他人详细解释如何进入~/test 目录时，即“输入cd，空格，波浪号，斜杠，test，斜杠，按回车键”。因为这里波浪号对应键盘上的按键，而不是绝对路径。关于路径的内容，我们也会在下一章做更详细的介绍。

命令提示符的最后一个部分是当前用户权限。Linux的系统权限状态很简单，即root用户和非root用户。处于root用户权限时，你可以执行任何操作，而非root用户只能执行有限的操作。不过现代的Linux系统中都拥有sudo命令，通过这个命令，可以使你在不切换到root用户的情况下，以root用户权限执行命令。该命令虽然对Linux权限模型做了点调整，但是仍然符合上述权限策略。

在命令提示符中，美元符表示普通用户权限。当我们以pi用户（非root用户）登录树莓派时，命令提示符最后是美元符号（$）。如果我们以root用户登录，我们会看到命令提示符最后是井号（#），例如：

```
root@raspberrypi ~ #
```

井号可以提示（警告）你当前处于 root 用户权限，并且在命令提示符中还会显示当前用户为 root，提醒你执行操作时一定要小心。每当你按下回车键之前，一定要注意用户名和权限提示。

命令提示符的内容也是可以修改的，有时系统管理员出于一些原因，会做出调整。虽然大多数人不会去修改命令提示符，不过当你接入一台 Linux 计算机，却看到了不同的命令提示符时，也不用紧张，各种命令行的操作还是一样的。

3.4 不同的 Shell

不同的终端操作大致相同，差异只体现在一些细节上，例如辅助功能，不过许多时候你可能并不会注意到这些差别。但在很多时候，你会察觉所连接的远程计算机使用的是不同的 Shell。Shell 不像浏览网络或者阅读 E-mail 的应用程序那么简单，它为操作系统的使用提供了完整的命令行环境，就像它的名字一样[1]，Shell 就是包裹在操作系统内核之外的花生壳。

目前为止，我们使用的都是 BASH（Bourne Again Shell）。BASH 是迄今为止使用最广泛的 Shell，每个 Linux 发行版中，都带有 BASH，并且常常作为默认 shell 提供。

不过在开源软件的世界，提供多种选择是必须的，所以 shell 也有不少选择。有的仅仅提供基础的功能，有的（例如 BASH）则提供了强大的功能。虽然 BASH 是主流的 Shell，不过当你不断地傲游 Linux 世界后，你还会遇到其他的 Shell。好在学会一个就可以有备无患。

我们不准备在本书中介绍其他的 Shell，因为即使你登录的计算机使用的是不同的 Shell，你也可以手动启动 BASH，回到你熟悉的环境（几乎所有的 Linux 系统都安装了 BASH）。而且其实系统命令在任何 Shell 中都是一样的，所以即使不使用 BASH，你也可以像之前一样操作计算机。事实上，很多不同的 Shell 从交互的命令提示符看来其实都是一样的，也许你都不知道你使用的不是 BASH。

本章小结

本章介绍了终端和它复杂的历史，包括为什么终端一直存活至今，以及它如何适应现代计算机的环境。之后我们学习了进入树莓派终端的 3 种主要方法：控制台、虚拟终端和 SSH 远程连接。本章还介绍了命令提示符的要点和注意事项。本章最后说明了不同的 Shell 会如何对你的工作、学习造成影响（好像几乎没有）。

下一章我们将和你一起探索树莓派，创建、修改和删除文件，还会做一些复杂的操作。我们将真正开始使用命令行，并学习一些实用的技巧。

[1] Shell，直译为外壳。

第4章

玩转文件系统

本章的重点将是文件系统，我们会一起了解操作系统如何管理它，介绍多个根目录文件系统和单一根目录文件系统的区别。

之后，我们将探索树莓派的文件系统，学习和了解 UNIX 系统的基本目录结构以及这些目录的作用。我们还会介绍 UNIX 系统“万物皆文件”（everything-is-a-file）的思想及其优点。

我们会学习如何管理文件系统，了解什么是绝对路经，什么是相对路径，如何创建目录和文件（也包括拷贝和删除）。我们还会学习 Linux 的文件权限、用户和用户组，如何通过它们管理文件访问，如何合理地控制文件权限。

最后，我们还要学习如何在 Linux 中创建类似快捷方式的文件。

4.1 什么是文件系统

我们试图按照字典上的定义向你解释什么是文件系统，但是最终我们发现，很多定义仅仅是“管理文件的系统”。因此这里还是直接按照我们的理解来定义文件系统，所谓文件系统是：

一种存储和组织计算机数据的方法，它使得对数据的访问和查找变得容易。

这个定义涵盖了计算机领域的文件系统，也包括了手机组织联系人的方式，甚至杜威十进制图书分类法也符合这个定义。

图书馆就是文件系统最好的例子，大部分图书馆都同时使用了至少两种图书分类方法（类似于文件系统）。例如小说类书籍一般按照作者的姓氏进行排序，而非小说类书籍则按照杜威十进制图书分类法排序。同时使用这两种系统，可以满足不同读者的需求。对于小说，很多人喜欢一些特定的作者，所以按照作者姓氏查找就很容易。而对于非小说类书籍，人们通常按照图书的主题进行选择，杜威分类法就是为这种情况而设计的。如果按照作者姓氏排序，估计你很难找到合适的书籍。

有些图书馆则将该系统更进一步发展，例如对于小说，先按照类型进行分类（科幻、幻想），之后再按照作者姓氏排序。这种方案下，你仍然可以按照作者姓名直接找到小说，因为同一作者的小说通常都是同一类型的，你同时还可以浏览类似的小说，扩大阅读面。

从图书馆的例子可以看出，合适的管理系统会使你对数据（书籍）的管理事半功倍。那么计算机世界的情况是怎样的呢？计算机的主要任务之一就是处理信息，所以需要一种方法，方便地

读取和存储信息。以前的计算机使用的是基于字母排序的系统，但当文件越来越多，该方案变得混乱且难以管理。

解决方法就是在文件系统中创建目录（或者也被称为文件夹），之后将相关的文件放入相应的目录。但是随着数据量的增长，目录方案还是会出现问题，你将会面对大量的目录，或是一个目录中大量的文件。

最终的解决方法是让文件系统支持创建多层目录，这样你可以按照你的需要组织文件和数据，并能容易地找到它们。加上现代的搜索技术，查找数据也变得更加容易。

多种文件系统

文件系统都是基于某种存储设备而存在的，包括硬盘、U 盘、DVD 光盘等等，这些设备可以独立地存储文件，因此你可以在不同的电脑之间读取设备上的所有数据。

每个文件系统都有根目录，根目录相当于设备的主目录，存储了所有的文件和子目录。你可以将文件系统理解为一棵树，每个文件和目录都是这颗树的一个树枝。当你在一台计算机上连接了多个文件系统，该如何将它们合理地呈现给用户？这时，出现了两种解决方案，第一种就是显示多个根目录，而另一种则显示单个统一的根目录。

多个根目录文件系统

这个方案由微软在开发 MS-DOS 时提出，并且一直沿用至今。每个设备都有自己的根目录，通过不同的根目录浏览不同设备上的文件。在 Windows 操作系统中，每个设备的根目录都分配了一个盘符。由于一些历史原因，系统盘符被称为 C:或者 C 盘，软驱是 A:和 B:（计算机曾经有两个软驱），如果没有其他的硬盘，光驱会被分配为 D:。

这个文件系统使用简单，通过盘符可以明确地区别不同的设备，不过这种方案也有一些缺点。

- 首先，由于使用英文字母作为盘符，26 个字母成为了障碍，你只能在计算机上安装最多 26 个文件设备。虽然对于家庭用户来说这不是什么问题，但是对于使用大型机的商业用户，需要安装数百个磁盘时，这种盘符方案就出现了问题。
- 其次，文件的路径受到设备物理位置的影响，如果你在计算机上连接了一个移动硬盘，并且将文件移动到其上面，那么文件的路径就会完全改变，这使其他人很难找到这个文件。

总的来说，Windows 使用的这种方案对于普通用户来说简单易用，但是对于大型分布式系统的管理员来说，却是很头疼的。目前，微软已经提供了一些技术来解决遇到的问题，例如基于网络共享技术的 Microsoft Distributed File System（DFS）。不过在 UNIX 系统中，采用了一种完全不同的文件系统方案来解决这些问题，即单个统一的根目录。

单个根目录文件系统

Linux 系统基于 UNIX 发展而来，使用单个根目录文件系统方案。因此 Linux 系统不像 Windows 系统为每个设备分配一个独立的根目录，而是只能指定一个设备挂载到/路径，也就是

UNIX 系统上唯一的根目录。在单个根目录文件系统中，Linux 是如何管理多个设备的文件系统的呢？

解决方法是将设备的文件系统映射到当前已经存在的某个目录中，也就是所谓的挂载文件系统，这样就可以挂载任意数量的设备到 Linux 系统上，你甚至可以将第二个移动硬盘挂载到第一个硬盘的目录中。对于用户来说，整个文件系统使用起来就像一个很大的磁盘，可以轻松地管理其中的文件，而不需要关注实际对应的物理设备。

文件系统挂载方案的优点在于使用时不用关注设备的物理存储方式，你可以将网络共享文件或者软件虚拟磁盘作为设备挂载到目录中。单个根目录文件系统缺点在于很难区分不同的设备，不过随着 Linux 系统的发展，这个问题也逐渐得到解决，尤其是现在图形化界面的广泛使用，区分不同的设备变得更加容易。

Mac 有所不同

Mac 计算机的 OS X 操作系统在解决这个问题方面有所不同，虽然 OS X 基于的是 UNIX 系统，实际使用的也是单个根目录的文件系统，但是除非你使用命令行，否则很难看出这一点。当你在 Mac 计算机上插入一个 U 盘时，OS X 系统和 Linux 系统类似地将其挂载在一个目录下，但是在系统界面中却看似 Windows，你可以独立地访问 U 盘的根目录，不过 Mac 并不使用盘符来标识设备，而是使用设备名称。

小结

上面的内容似乎有些理论化，不过这对于我们后面学习命令行操作有很大的用处。当你第一次接触 Linux 系统时，由于它和 Windows 系统的巨大不同，会给你对于系统的理解造成较大的障碍。记得 15 年前，第一次安装 Linux 系统时，“you need a root / device”困扰了我们很久。搞明白路径和物理设备的关系，才能保证不会误删除文件，才能保证不会做出错误的操作。

希望上面的介绍能为你后面的阅读带来帮助，帮助你理解 Linux、Windows 和 Mac 在文件系统管理方面的不同。

4.2　万物皆文件

Linux 处理磁盘分区的方式很好地体现了万物皆文件的思想，但是树莓派没有磁盘（将 SD 卡当作磁盘使用），更没有分区，所以我们没法以树莓派为例子向你展示这一点，那么来看看我们的服务器。

```
/dev/sda
/dev/sda1
/dev/sda2
/dev/sdb
/dev/sdb1
```

可以看到，所有的磁盘分区都以文件的形式存放在/dev 目录下，该目录是 Linux 的设备文件目录，我们后面还会详细解释这个目录的作用。我们先来看一下这些设备文件，你会注意到所有文件都是以 sd 开头的，因为这些都是 SCSI 磁盘。如果使用的是 IDE 磁盘，那么你会看到以 hd 为开头的设备文件名称。

目前我们知道计算机上安装了几个 SCSI 磁盘（SATA 磁盘），我们从这些设备文件上还能得到什么其他的信息么？首先，我们注意到设备文件有 sda 和 sdb 两种，这是因为 Linux 系统将第一块 SCSI 磁盘命名为 sda，将第二块命名为 sdb，当然如果安装了第三块就是 sdc，所以我们能知道计算机上实际安装了两块磁盘。

到这里，我们确切地知道了计算机上安装了几块磁盘，那么设备文件名称最后的数字是什么意思呢？在这个例子中，这些数字表示的是磁盘上的分区，第一个分区就是以 1 结尾的，第二个分区就是以 2 结尾的，以此类推。所以我们能够通过设备文件名称知道计算机上安装的磁盘数量，还能了解每个磁盘上的分区情况。

你可能想知道哪个磁盘是当前的系统盘呢，但实际上它们都已经挂载到了系统之上，我们更需要关心的是如何访问它们。万物皆文件的设计下，磁盘的分区也是以文件的形式呈现出来的。我们需要访问磁盘或者分区的内容时，可以通过设备关联文件来进行操作。这种通用的设计使得在 Linux 系统上，所有的资源都可以以统一的方式进行访问，这使得开发程序和使用系统变得更加容易，后面我们还会在第 5 章详细介绍这一特点。

我们通过一个实际的例子来介绍设备文件的使用。比如你想修改磁盘的分区，你将会使用 fdisk 工具，fdisk 需要知道具体的设备路径，这里我们将会修改第二块磁盘，那么我们使用以下的命令：

```
$ fdisk /dev/sdb
```

因为我们想修改的是磁盘中的分区状态，所以我们提供了磁盘设备的路径/dev/sdb，而不是某个分区。分区完成后，我们还要格式化某个分区，以供 Linux 使用。我们将使用 mkfs.ext4 工具，这里我们将会提供格式化的分区的路径，例如我们想格式化第一个分区，那么命令如下：

```
$ sudo mkfs.ext4 /dev/sdb1
```

这条命令很明确，我们格式化了第二块磁盘的第一个分区，其实 mkfs.ext4 命令并不关心你提供的路径是否是一个分区，如果你没有输入最后的 1，那么将会格式化整个磁盘。由于万物皆文件，所有的设备操作起来都是一个文件，很多工具只是在操作文件，它们并不关心路径对应的物理设备是什么。

万物皆文件的设计之下，Linux 系统管理了所有设备，为用户和应用程序开发人员提供了基于文件的访问方法，隐藏了各种硬件设备的不同特点和不同操作方法。实际上/dev 设备目录中的文件并不真实存放于某个物理位置，它们是 Linux 系统为所有设备创建的虚拟文件系统。说到树莓派，由于并不存在任何 SCSI 或者 SATA 磁盘，所以你不会找到任何/dev/sd 文件。

4.3　文件系统结构

本节我们介绍树莓派文件系统的结构，它和标准的 Linux 系统很类似，所以你不仅能学习树莓派的知识，还能同时了解了 Linux 系统。

/（根目录）

根目录（这里并不是指 root 用户的主目录）表示文件系统的最顶层，所有的文件和子目录都在根目录之下。我们之前介绍过，Linux 系统使用单个根目录的文件系统，所以根目录是所有路径的起点。

/root

该目录是 root 用户的主目录，它很容易和根目录（/）混淆，尤其是你正在使用 root 用户的时候。所以当你使用/root 时，请和根目录区分清楚。

/etc

/etc 目录是系统最重要的目录之一，几乎所有的系统和应用程序（比如 Apache Web 服务器）的配置文件都存放在这里，很多 Linux 新手备份应用时都会忘记备份/etc 中的配置文件。请谨慎对待该目录中的文件，不要忘记经常备份。

/proc

/proc 目录是一个虚拟文件系统，Linux 内核通过该目录向用户空间的应用程序提供系统资源的访问能力，包括系统的状态和正在运行的所有进程。该目录中最常用的两个虚拟文件分别是 CPU 配置文件（路径为/proc/cpuinfo）和内存使用状态文件（路径为/proc/meminfo）。该目录中的文件都是虚拟的，而且大多数都是只读的。不过也有一些文件可以写入，为你提供了动态调整内核参数和系统设置的能力。

/var

/var 目录一般用于存放系统和应用程序创建的运行文件，例如对于大多数应用程序，日志文件都存放在/var/log/目录，文件锁都存放在/var/run/目录。文件锁中不存放任何数据，应用程序仅仅用它们来表示某个运行状态，保证自己不会同时运行两次。另外，在树莓派中，Apache web 服务器还使用/var/www/目录存放网站的文件。有些 Linux 发行版中，这些文件被存放于/srv/目录。不过/var 目录仍然是最主要的运行文件存放目录。

/boot

该目录的作用是存放系统的启动文件，包括启动器（bootloader）和 Linux 内核。以前的计算机上，/boot 目录使用一个单独的小分区，因为当时的计算机还不能直接从一个很大的分区启动。不过现代的计算机已经没有这种问题了，所以/boot 可以和根目录使用相同的分区。

/bin和/sbin

这两个目录存放了各种程序文件，当你在命令行中输入命令运行程序时，Linux 系统会在一

些指定的路径中查找应用程序，这些路径（默认包括/bin 和/sbin）组成了 PATH 环境变量。一般情况下，普通用户的 PATH 环境变量中只有/bin 目录，不需要也不存在/sbin 目录。/sbin 用于存放一些系统管理程序，普通用户很少用到。

/dev

之前“万物皆文件”一节介绍过，该目录存放了所有设备的映射文件，并且为系统工具访问硬件提供了方便。除了我们之前曾经讨论过的磁盘设备文件，还包括了显卡、声卡、虚拟终端等设备文件。

注意 /dev 目录中没有网卡设备文件，因为它们的使用方法比较特殊。如果需要了解网卡信息，可以使用 sudo ifconfig –a 命令，该命令将会列出所有网络设备的状态。详细了解该命令，请参考第 3 章和第 9 章。

/home

一般情况下，/home 中存放了所有用户的主目录，并且每个用户的主目录都对应独立的分区、磁盘或者网络共享文件，这样设计的目的是为了分割不同用户的系统和应用数据，使多个用户可以互不干扰地使用系统。大多数基于 UNIX 设计的操作系统都这样存放用户的主目录，不过也有些用户主目录存放在其他路径（root 用户的主目录），比如在 Mac 计算机上，用户主目录存放在/Users 中，还有些商业公司使用自定义的位置来存放用户主目录。

/lib

该目录存放了应用程序使用的库文件。库文件的作用是将应用程序需要的函数集合在一起，并且能够在不同的应用程序之间共享以节省空间。例如很多应用程序使用相同的数据库链接库来访问数据库，而不需要每个应用程序都自带一份。为了让所有应用程序都能找到所需的库文件，需要将这些文件存放在一个固定的路径中，比如/lib。所以如果你需要修改该目录中的文件，请一定小心，因为任何修改都可能会影响系统的运行。

/lost+found

这个目录比较特别，本质上它不能算是 Linux 文件系统的一部分。文件系统无法记录某个文件时，它会将其存放在该目录中。比如磁盘出现了错误或者损坏，你尝试进行磁盘恢复，有些恢复出来的文件已经无法确定原来的存放路径，Linux 系统就会将其存放在这个特殊的目录中。所以，如果不出现什么意外的话，你完全不会去查看这个目录的内容。

/mnt

该目录的名称是 mount 单词的缩写，一般情况下，该目录用来进行文件系统的挂载。比如你准备连接一个网络共享目录或者外置硬盘，你可以在/mnt 中创建一个目录，并将它们挂载在那里。软盘和光盘也被挂载在该目录下，一般情况下软盘挂载路径是/mnt/floppy，光盘挂载路径是/mnt/cdrom。不过，随着计算机使用的发展，人们更多地开始使用移动媒体，它们通常都不挂载在/mnt 中（参考下面的/media），该目录的使用也越来越少。

/media

该目录是近年来才被加入 Linux 文件系统的，独立的/media 目录是为了明确地区分可移动媒体（例如 U 盘、相机或者多媒体播放器）和挂载设备（挂载在/mnt 目录）。该目录中的内容一般由 Linux 自动挂载系统管理，你可能很少有机会直接操作该目录。

/usr

该目录存放了计算机上安装的几乎所有软件，所以一般也是系统上最大的一个目录（如果不考虑有很多用户的主目录）。各种软件需要的文件都会存放在该目录中，一般也无需你手动管理。

/opt

该目录没有明确的使用定义，有的系统用它来存放应用，有的系统则根本不使用该目录。有些第三方应用会使用该目录存放文件，例如 Oracle 数据库服务就是默认安装在/opt/中的。你可能很少接触该目录，不过在系统备份时请不要忘记该目录中的文件。

/srv

该目录也是最近才被加入文件系统的，它被设计为存放系统服务的运行文件。不过目前不是所有服务程序都会使用该目录，如果你发现以前存在于/var/中的一些文件不见了，可以在该目录中找找看。

/sys

该目录存放了一些系统信息，不同于/proc/中的虚拟文件（存储于内存中），该目录中的文件是真实存放于磁盘上的，该目录似乎也很少用到。[1]

/tmp

顾名思义，该目录存放了各种应用程序创建的临时文件。默认情况下，该目录中的文件每次重启后都会清空，所以这里仅仅是个临时空间，使用时请注意这一特点。

小结

在这一节，我们介绍了 Linux 文件系统中的一些重要目录，其他目录基本都是这些目录的子目录。不过也不是所有系统都遵循这些命名规则，可能会使用其他名称来替代它们。

4.4 让我们来实践一下

我们已经学习了很多理论知识，包括文件系统是如何工作的，UNIX 系统（Linux 系统）是如何管理单个根目录文件系统的。我们介绍了“万物皆文件”的设计思想，了解了其中的特殊目

[1] 译者注：在较新的 Linux 发行版中，/sys 都是作为 sysfs 的挂载点存在，而非真实文件。

录和虚拟文件。最后，我们还介绍了文件系统中各个主要目录的作用。不过到现在，我们还没有尝试操作一下文件。

所以本章后面的内容，我们将一起学习如何使用文件系统，包括如何执行创建目录、移动文件等操作。阅读完后面几节，你将能够掌握文件、目录的创建和删除，还会了解一下 Linux 文件权限，以及如何进行正确的设置。那么就让我们开始吧。

使用 pwd 命令显示当前目录路径

我们第一个介绍的命令功能是了解当前的目录路径，让我们打开命令提示符，就像上一章中介绍的，你会看到我们正处于主目录中。

```
pi@raspberrypi: ~$
```

我们已经学习过，波浪线表示主目录。但是如何知道当前的目录路径呢？我们可以执行 pwd 命令，虽然我们上一章已经介绍过了，不过我们还是再看一下。

```
pi@raspberrypi: ~$ pwd
/home/pi
pi@raspberrypi ~$
```

该命令可以告诉我们当前的路径，但是其实信息还是不够，你只是知道在哪里，但是当前目录是什么状态还是无法得知。就像你在一个房子里，你知道了地址，但是你还需要了解房子中有什么。此时，需要执行的就是 ls 命令。该命令有很多的参数，不过我们只要记住最简单的用法就足以满足日常的使用了。

使用 ls 命令显示目录内容

我们先看一个例子：

```
pi@raspberrypi: ~$ ls
Desktop    python_games
pi@raspberrypi: ~$
```

书中可能无法显示出颜色，但是我们看到 Desktop 和 python_games 是以深蓝色显示的（你也应该可以在你自己的树莓派中看到类似的效果）。深蓝色的名称提示我们这两个是目录而不是文件。看起来似乎没有文件处于当前目录中，这是由于该目录中的文件都是隐藏的。在 Linux 系统中，文件名以.开头的都被认为是隐藏文件。这些隐藏文件内容也没什么特别，大部分都是应用程序的配置文件或者临时文件。ls 命令遵照了这一命名惯例，默认不显示隐藏文件。不过如果你的确需要看到隐藏文件，可以为 ls 命令加上-a 参数，就能看到类似如下的结果。

```
pi@raspberrypi: ~$ ls -a
.              .cache
..             .config
.bash_history  .dbus
.bash_logout   Desktop
.bashrc        python_games
.dbus          .ssh

pi@raspberrypi: ~$
```

使用 touch 命令创建空文件

下面，让我们来创建自己的文件，并进行操作。目前，我们尚未涉及创建以及编辑文本文件等相关内容（相关内容请阅读第 6 章），这里先介绍一种创建文件用的小工具 touch。Linux 中每个文件都具有两个时间戳：

- 创建时间戳
- 最近修改时间戳

通过这两个时间戳我们可以了解一个文件的创建和修改的情况。对系统管理员来说这类信息自然十分实用，通过它们可以方便地分辨出哪些文件被频繁使用。诸如文件备份脚本等工具就需要利用这类信息来决定需要对哪些文件执行备份操作。在某些情况下，我们可能需要在不改变文件内容的情况下更新时间戳，这时就需要借助 touch 了。touch 命令可以直接更新文件的时间戳。然而，如果你指定的文件名不存在的话，touch 会以该文件名创建一个空文件。所以，我们可以使用 touch 方便地创建空文件。好了，让我们来试着创建一些文件吧。

```
pi@raspberrypi: ~$ touch raspberry
pi@raspberrypi: ~$ touch pi
pi@raspberrypi: ~$ ls
Desktop pi python_games raspberry
pi@raspberrypi: ~$
```

可以看到，我们使用 touch 命令创建了两个新文件。两个新文件以灰色显示，这是由于 ls 命令通过不同颜色显示不同类型的文件，这样方便我们对文件进行管理。例如，所有可执行文件会显示为绿色，稍后我们会进行更详细的介绍。

当我们需要处理的文件有几十、上百个时，显然通过目录进行管理会方便得多。下面我们来介绍如何在命令行下新建目录。

注意　目录（directory）和文件夹（folder）这两个词通常表示相同的东西。目录一词出现较早，而后微软公司开始使用文件夹这一更容易理解的概念。随着更多的 Windows 用户迁移到 Linux，在 Linux 世界中，文件夹一词的使用也开始日益频繁了。

使用 mkdir 创建目录

我们可以方便地通过 mkdir 命令创建目录。值得注意的是，如果已存在同名的文件或目录，mkdir 命令会给出相应的出错提示。这里我们创建一个名为 pifun 的目录。

```
pi@raspberrypi: ~$ mkdir pifun
pi@raspberrypi: ~$ mkdir pifun
mkdir: cannot create directory 'pifun': File exists
pi@raspberrypi: ~$
```

可以看到第二次试图创建 pifun 目录时 mkdir 命令给出了出错提示。出错信息中的“File”泛指文件和目录。让我们再来回顾下当前目录里的内容。

```
pi@raspberrypi: ~$ ls
Desktop pi pifun python_games raspberry
pi@raspberrypi: ~$
```

使用 mv 命令

好了，让我们来整理下我们创建的这些内容吧。通过 mv（move 的缩写，移动）命令我们可以将文件移动到刚才创建的 pifun 目录中。与之前介绍的命令稍有不同，mv 命令有两个参数。你需要告诉 mv 命令需要移动些什么以及移动到哪儿。与大多数命令相同，mv 命令的第一个参数为源文件（source），第二个参数为目标文件（destination）。实际操作方式如下：

```
pi@raspberrypi: ~$ mv pi pifun
pi@raspberrypi: ~$ mv raspberry pifun
pi@raspberrypi: ~$ ls
Desktop pifun python_games
pi@raspberrypi: ~$
```

让我们来检查下 mv 命令是否正常工作了。你可以通过 cd 命令切换到 pifun 目录，然后通过 ls 命令来验证，也可以直接通过 ls pifun 来查看 pifun 目录中的内容，如下所示。

```
pi@raspberrypi: ~$ ls pifun
pi raspberry
pi@raspberrypi: ~$
```

使用 cp 命令来复制文件

对新手系统管理员来说，花费大量时间移动各种文件、查找各种文件的位置是难以避免的。我们已经知道可以通过 mv 命令移动文件，那么如何复制文件呢？使用 cp（copy 的缩写，复制）命令我们可以方便地实现这样的操作。

```
pi@raspberrypi: ~$ cd pifun
[pi@raspberrypi pifun]$ cp pi pi2
[pi@raspberrypi pifun]$ ls
pi pi2 raspberry
[pi@raspberrypi pifun]$
```

现在我们有了 pi 文件以及其复本 pi2，下面我们再尝试复制目录。

```
[pi@raspberrypi pifun]$ mkdir moarpi
[pi@raspberrypi pifun]$ cp moarpi moarpi2
cp: ommiting directory 'moarpi'
[pi@raspberrypi pifun]$
```

可以看到，命令没有像我们期望的那样工作。这是因为 cp 命令默认只能复制独立的文件而不能复制整个目录。这样设定的原因在于，复制目录的操作可能会耗费大量的系统资源，有必要和复制单个文件进行区分。

那么当我们确实需要复制目录时该如何操作呢？在进行备份时，显然我们需要进行这样的操作。这时，我们需要通过 –r 选项告知 cp 命令我们想要进行递归的（recursively）复制。